

Computational Design of Skinned Quad-Robots

Xudong Feng, Jiafeng Liu, Huamin Wang, *Member, IEEE* Yin Yang, *Member, IEEE* Hujun Bao, Bernd Bickel, and Weiwei Xu, *Member, IEEE*

Abstract—We present a computational design system that assists users to model, optimize, and fabricate quad-robots with soft skins. Our system addresses the challenging task of predicting their physical behavior by fully integrating the multibody dynamics of the mechanical skeleton and the elastic behavior of the soft skin. The developed motion control strategy uses an alternating optimization scheme to avoid expensive full space time-optimization, interleaving space-time optimization for the skeleton and frame-by-frame optimization for the full dynamics. The output are motor torques to drive the robot to achieve a user prescribed motion trajectory. We also provide a collection of convenient engineering tools and empirical manufacturing guidance to support the fabrication of the designed quad-robot. We validate the feasibility of designs generated with our system through physics simulations and with a physically-fabricated prototype.

Index Terms—Computational Fabrication, Motion Design, 3D-Printing, Physics-based Simulation.

1 INTRODUCTION

The design and construction of robots that can execute compelling motions is a challenging task. It requires careful geometric planning of robotic mechanisms and professional knowledge of the kinematic and dynamic behavior of the robot. Embedding such knowledge into procedures of computational robot design [1], [2] in conjunction with rapid prototyping techniques, such as 3D printing technology, bears tremendous potential to accelerate the construction of personalized robots. For instance, Megaro et al. [3] used a kinematic optimization algorithm for the design of multilegged robots consisting of rigid links. However, real-world creatures are not merely rigid skeleton rigs. The muscle and flesh surrounding the skeleton provides diverse morphologies, enriched expressivity. For instance, people wearing a prosthetic limb often prefer a highly realistic rubbery artificial arm over a more functional mechanical one [4]. Moreover, skins and muscles might be essential for facilitating challenging tasks, such as reproducing compliant grasping of a hand or swimming motions of a fish [5], [6].

Different to the simulation of the robots with rigid links only, the influence of the soft body on the control torques at actuators and on the contact forces at the ground should be simulated and fully taken care of to judge the plausibility of a designed soft robot. This imposes a significant computational challenge for the motion design and fabrication of skinned robots. In this paper, our goal is to address this problem by integrating both dynamic simulation and kinematic optimization into the motion design of quad-robot systems, with soft skins attached as their organic embodiments. Its kernel is an optimization problem that integrates both user-provided kinematic preference and physical constraints of the robot to obtain a dynamically feasible motion plan. The primary physical constraint is the dynamic viability of the skeleton trajectory when

a soft skin is attached. The dynamics of the robot is formulated as two-way coupled subsystems of the rigid multibody system and the deformable skin. In addition, our system also incorporates motor constraints and a stability constraint. The motor constraints ensure that the calculated joint torques are within the physical limits of the installed servo motors, whereas the stability constraint requires the center of projection (COP) of the robot structure to fall inside the supporting polygon. As a result, given the surface mesh and desired kinematic trajectories of the robot, our pipeline generates a physically-valid motion plan that can be realized under the given hardware constraints. The robot itself can be fabricated using rapid prototyping technology.

While space-time optimization is widely used in long-horizon motion design problem, it is computationally prohibitive if directly applied to our case of two-way coupled system, mostly due to the large number of DOFs of the soft skin mesh coupled with the skeleton and various physical constraints. To this end, we propose an alternating optimization algorithm of two optimization steps to mitigate the computational cost: (1) Space-time optimization with respect to the DOFs of the rigid skeleton while assuming the deformation of the soft skin remains the same as the previous iteration. (2) Frame-by-frame optimization as in [7] to further optimize the motion plan obtained in step 1 according to all the physical constraints at each frame. To this end, we non-trivially extend the spring-based control force formulation in [7] to handle the full simulation of the two-way coupled rigid skeleton and soft skin. Our solver can efficiently handle the Lagrange multipliers introduced by the coupling constraints and the collisions between the skeleton and skin. With a proper initialization, these two steps are alternatively executed until the convergence. In the space-time optimization step, the influence of the soft skin to the rigid skeleton is treated as known quantity and simplified to be the coupling forces and the influence of the center of mass (COM) at each link in the skeleton due to the skin deformation. This setting makes the space-time optimization computationally efficient through decoupling the skin mesh DOFs, which is critical to achieve global effect in the motion design. The skin deformation is updated after each frame-by-frame optimization.

To ease fabrication, we provide a convenient workflow with tailored engineering tools and empirical fabrication guidance

- Xudong Feng, Jiafeng Liu, Hujun Bao and Weiwei Xu are with State Key Lab of CAD& CG, Department of Computer Science, Zhejiang university, Zhejiang 310058, China. Weiwei Xu is the corresponding author. E-mail: {bao, xww}@cad.zju.edu.cn
- Huamin Wang is with Department of Computer Science and Engineering, Ohio State University. E-mail: whmin@cse.ohio-state.edu.
- Yin Yang are with the Department of Electrical and Computer Engineering, University of New Mexico. E-mail: yangy@unm.edu.
- Bernd Bickel are with Institute of Science and Technology Austria, Vienna, Austria. E-mail: bernd.bickel@gmail.com.

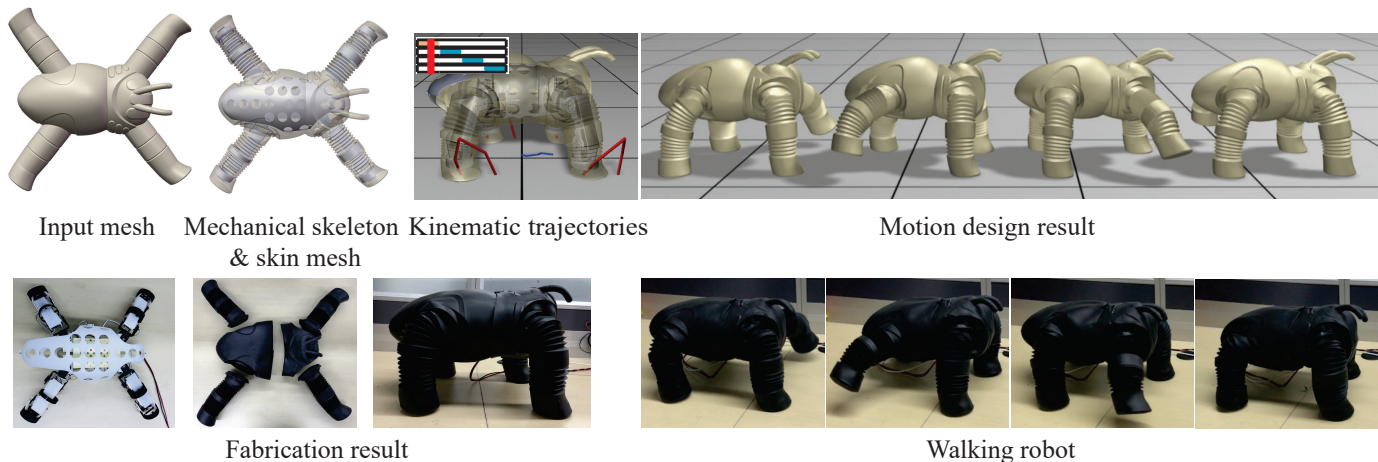


Fig. 1. We propose a computational fabrication system for designing and fabricating skinned quad-robots. Given an input mesh representing the shape of a quad-robot such as this beetle-like robot, we design its mechanical skeleton with motors to drive its locomotion. The motion plan is generated by an optimization algorithm with kinematic trajectories as the user input. The trajectories consist of the foot swing trajectories (red curves), center of mass (COM) trajectory (blue curve) and foot contact plan (yellow bars) input by the user. Our optimization fully takes account of all the physical and dynamical constraints. By fabricating this robot design, it can be verified that our algorithm is able to generate plausible and physically feasible motion plans for quad-robots, and the simulated results well match the physical experimental results. Note that we only render the transparent input surface without thickness to demonstrate the surface-structure coupling geometry.

embedded in a standard CAD software, empowering regular users to design quad-robots. The modular design scheme allows the user to quickly start from a design template of the mechanical skeleton in *SolidWorks* and adapt it to the body shape. The rigid skeleton is fabricated via 3D printing, and the skin is separately fabricated using injection molding by pieces. We tested the optimization algorithm on skinned quad-robots with varying body-to-leg ratios and different mechanical skeletons. Both physical and numerical experiments show that the proposed algorithm is an effective means of obtaining physically valid motions of the skinned quad-robots.

2 RELATED WORK

Computational fabrication aims at designing and creating physical artifacts with the help of computational methods. A large class of methods addresses inverse design problems by incorporating fabrication limitations in geometric design algorithms via constrained optimization or the integration of fast simulation techniques [8], [9]. This line of research enables the design of objects with a wide range of controllable physical and mechanical properties, such as appearance [10], [11], [12], [13], deformation [14], [15], [16], [17], articulation [18], [19], [20], and mechanical motion [21], [22], [23], [24], [25]. Some existing contributions also investigated how to instantiate virtual characters as 3D-printable physical entities like mechanical robots [3], [21]. Yet, these methods merely focus on robots consisting of rigid links and basic balancing constraints and/or velocity limits.

Bickel et al. [26] proposed a process for designing synthetic skin and actuation parameters for animatronic characters that mimic facial expressions of a given subject. Skouras et al. [15] optimized the internal material distribution so that the resulting character exhibits the desired deformation behavior. Focusing on actuation, Bern et al. [27] computed the layout of winch-tendon networks to animate plush toys, and Ma et al. [28] optimized the chamber structure and material distribution for designing soft pneumatic objects.

Our work shares some of these goals but takes a significantly different approach. Instead of relying on a rigid or quasi-static

underlying simulation of the skin deformation in our case, the influence of the soft skin on the embedded moving mechanical skeleton makes us face a dynamic two-way coupled multibody-elastic problem. It is much more challenging to accurately simulate and optimize due to the drastically increased system complexity, nonlinearity and discontinuity, i.e., due to the complimentary constraints enforced at the contact vertices.

Physics-based character motion generation has vast applications in both graphics and robotics. Algorithmic approaches include leveraging space-time optimization with necessary physical constraints and developing controllers to drive forward simulations.

The seminal work by Witkin and Kass [29] generated motion trajectories by optimizing physical constraints and animator controls at key frames, a well-known space-time constraints framework for animation. With proper motion data, the space-time optimization produces realistic articulated motions for bipedal or multilegged characters through different physical properties [30], [31], [32], [33], [34], [35], [36], [37], [38]. It can be used to transform motion capture data into physically plausible motions [39].

The locomotion controller aims to compute joint torques or control forces to drive the locomotion behaviors of articulated figures. The joint torques are usually calculated via the proportional and derivative (PD) controller such that the rigid skeleton of a character follows designated joint angle trajectories [40], [41], [42]. Balance control strategies, such as the swing foot placement or zero moment point, are essential to generating stable locomotions. [40], [41], [43], [44], [45], [46], [47], [48]. Continuous adaptation of the target joint trajectory for balancing a walking human was developed in [49]. Controllers that produce highly dynamic skills for human animation were suggested in [50], [51], [52]. The joint torques can also be computed via optimal control to approximate the motion capture data or motion data from kinematic simulators [53], [54].

Our work is inspired by studies on how to drive the soft skin deformation with the underlying rigid skeletons or pseudo muscle force [7], [55], [56]. Two-way coupling of rigid bodies and elastic bodies was considered in [57]. Fast simulation and control of soft robots of various configurations and actuations has also been

studied using finite element method and the reduced formulation of compliance matrix [58], [59], [60], [61], [62].

Elastic body simulation focuses on the formulation of an elastic deformation energy and the proper handling of contact constraints to simulate realistic deformations of soft bodies [63], [64], [65], [66]. A comprehensive survey of physics-based elastic deformation models can be found in [67].

Space-time optimization techniques can also be applied to control the motion of elastic bodies that are represented by volumetric meshes. To reduce the number of variables used to control the vertex positions in the optimization, model reduction techniques are frequently used [68], [69], [70]. Barbič et al. [68] imposed the equation of motion constraint in elastic body deformation, using the discrete adjoint method to compute the gradients of control forces. Pan et al. [71] integrated the contact forces as additional variables to handle environment interactions and solved the space-time objective with alternating optimization, but did not handle the two-coupling problem we want to solve.

3 OVERVIEW

Given an input surface mesh of a quad-robot, we first design mechanical skeleton and skin mesh (Section 6.1) for the robot, and then use the proposed two-step alternating algorithm to optimize for a physically plausible motion plan (Section 5). The overall system flowchart is illustrated in Fig. 1.

During the alternating optimization, the first space-time optimization step outputs joint angle trajectories for the design according to the user-specified end-effector and COM trajectories (Section 5.1). This step is made possible by only considering the approximated skin deformation. The second frame-by-frame optimization step improves the physical plausibility of the joint angle trajectories with full simulation and various physical constraints (Section 5.2), such as physical torque limits for the selected motors in the design. Two-way coupled multibody-elastic dynamics (Section 4) are adopted in this step for the full simulation.

Finally, the designed robot is fabricated by fast prototyping methods for a physical validation (Section 6.2), and stepper motors are mounted to drive the skeleton and the attached soft skin to realize the motion plan.

4 TWO-WAY COUPLED MULTIBODY-ELASTIC DYNAMICS

A core ingredient of our system is the dynamic simulation of the robot using a self-actuated rigid skeleton with a soft skin attached. We are inspired by existing coupled simulation systems in graphics [72], [73], [74] and exploit the Lagrange multipliers method to enforce the two-way coupling between the skeleton and the soft skin, which can be naturally integrated into the subsequent locomotion optimization. The Lagrange multipliers are used to guarantee that the skeleton and skin are attached to each other at prescribed locations, and we solve for all the unknown DOFs from both subsystems simultaneously.

4.1 Two-Way Skeleton-Skin Coupling

We use Lagrangian mechanics for both the rigid skeleton and soft skin and obtain a symmetric formulation for these two subsystems. To ease the formulation of motion constraints, we use the generalized coordinates \mathbf{q} to parameterize the entire skeleton,

where $\mathbf{q} = \{c_x, c_y, c_z, q_1, \dots, q_m\}$. The vector \mathbf{q} is composed of the Cartesian coordinates of the center of mass (COM) at the root link $\{c_x, c_y, c_z\}$ and the three Euler angles at each joint $\{q_1, \dots, q_m\}$. We opt to model the soft skin using the neo-Hookean material model [75] because it has been demonstrated to be well suited for robotic skins made out of silicone [76] and can handle large local deformations induced by joint rotation observed in our examples. If necessary, however, our approach should be easily extensible to more sophisticated material models such as Mooney-Rivlin and Ogden. The DOFs representing the vertex displacements of the tetrahedral skin mesh are denoted by the vector \mathbf{u} . Though numerical discretization, the equations of the forward simulation of rigid skeleton and soft skin at each time step can be written into a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where \mathbf{A} is the system matrix and \mathbf{b} is a vector of the sum of constant terms and external forces. The detailed derivation of two linear systems, $\mathbf{A}_r, \mathbf{b}_r$ for rigid skeleton and $\mathbf{A}_d, \mathbf{b}_d$ for soft skin, are elaborated in Appendix A.

The two subsystems are coupled by attaching the soft skin to the underlying skeleton at prescribed *glue vertices*. Mathematically, this straightforward treatment leads to a set of nonlinear position constraints:

$$\mathcal{C}(\mathbf{q}, \mathbf{u}) = \mathcal{R}(\mathbf{q})\mathbf{r} + \mathbf{t} - \mathbf{x}_c = \mathbf{0}. \quad (1)$$

The matrix \mathcal{R} converts the positions on the rigid links where the skin is attached, \mathbf{r} , from local to world coordinates. This operation can be easily expressed as a rotation chain from the links all the way back to the root. Meanwhile, \mathbf{t} concatenates the root translations of the rigid links, and \mathbf{x}_c denotes the positions of the glue vertices on the skin mesh $\mathbf{x}_c = \mathbf{S}_c\mathbf{u}$, where \mathbf{S}_c is a selection matrix.

Using the Lagrange multipliers method, we obtain the coupled multibody-elastic system:

$$\begin{bmatrix} \mathbf{A}_r & \mathbf{0} & \nabla_{\mathbf{q}}\mathcal{C}^\top \\ \mathbf{0} & \mathbf{A}_d & \nabla_{\mathbf{u}}\mathcal{C}^\top \\ \nabla_{\mathbf{q}}\mathcal{C} & \nabla_{\mathbf{u}}\mathcal{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{q} \\ \Delta\mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_r \\ \mathbf{b}_d \\ \mathbf{0} \end{bmatrix}. \quad (2)$$

The coupling constraint is linearized via $\nabla\mathcal{C}$. In each time step, we solve for the changes of the system DOFs in Eq. 30. Thus, we have $\mathbf{q}^i = \mathbf{q}^{i-1} + \Delta\mathbf{q}$ and $\mathbf{u}^i = \mathbf{u}^{i-1} + \Delta\mathbf{u}$, where the superscript $[\cdot]^i$ indicates the frame index.

4.2 Collision and Contact Handling

There are two types of collisions/contacts we need to take care of in our simulation. The first is the collision between the robot's feet and the ground surface, which provides necessary support and friction forces to the robot. As will be detailed in Sec. 5.2, we resolve them using linear complementary constraints (LCP) to guarantee the physical feasibility of the locomotion.

The second is the self-collision of the soft skin; rotating joints tend to compress the inward skin and induce self-collisions. Moreover, skin-skeleton collisions¹ could also occur when the skeleton is being articulated. Because such collisions typically take place under low relative velocities, we handle them using the explicit penalty force method [77], [78].

5 MOTION PLAN OPTIMIZATION

As the system input, the trajectories of the COM, end effectors, and the footfall pattern are provided by the user. Our system

1. As discussed in Sec. 6, skin-skeleton collisions can occur between the skin and the bracing unit, not the mechanical skeleton.

generates a dynamically feasible motion plan for the robot that resembles as much as possible the one prescribed by the user. A motion plan, defined as $\mathcal{P} = \{\mathbf{q}^i, i = 1 \dots N; \Delta t\}$, includes the skeleton configuration \mathbf{q}^i of the i -th frame for $i = 1$ to N , and the time step size Δt .

Previous works, e.g. [Megaro et al 15], solved the motion design problem by simultaneously finding the optimal skeleton configuration for all the frames in \mathcal{P} . However, this problem becomes much more challenging in the case of a multibody-elastic system, due to the large number of DoFs and associated physical constraints. Thus, the global approach with full DOFs is infeasible in our case. In this section, we elaborate the details of our two-step alternating motion plan optimization algorithm which uses approximated skin deformation to significantly improve the efficiency of the global space-time optimization. Specifically, the influence of the skin deformation on the skeleton is approximated as the coupling forces at glue vertices and the influenced COM positions of each link. The convergence of the algorithm is tested considering the difference between the space-time optimized one and the optimized one after the frame-by-frame optimization step. Fig. 2 illustrates the algorithm flowchart.

5.1 Space-time Optimization With Skin Deformation Approximation

Similar to the formulation in [36], for each i -th frame, we consider the set of generalized coordinates of the rigid skeleton, \mathbf{q}^i , together with the contact forces $\mathcal{F}_{c,j}^i$ and torques $\boldsymbol{\tau}_c^i, j$ that are exerted on the j -th end effector in contact with the ground. The influence of the skin deformation to the skeleton at i -th frame is simplified to be the coupling forces $\boldsymbol{\lambda}^i$ at glue vertices and the skin deformation \mathbf{u}^i . They are obtained from the previous frame-by-frame optimization and not optimized in this step. The displacements in \mathbf{u}^i are represented into the local coordinate frames of links at each frame in order to compute the influence of the skin deformation to the COM of the robot.

Given these quantities, the optimization objective is defined as the weighted sum of six terms that balance the user-specified end effector and COM trajectories and the smoothness of the optimized motion:

$$E_A = \min \sum_i (\alpha_\tau E_\tau^i + \alpha_S E_S^i + \alpha_C E_{COM}^i + \alpha_e E_{EE}^i + \alpha_f E_F^i + \alpha_o E_O^i). \quad (3)$$

The first term E_τ^i is standard in space-time optimization to minimize the torques $\boldsymbol{\tau}^i$ exerted at the joints:

$$E_\tau^i = \frac{1}{m} \|\boldsymbol{\tau}^i(\mathbf{q}^i, \boldsymbol{\lambda}^i, \mathcal{F}_c^i, \boldsymbol{\tau}_c^i)\|^2.$$

The torques $\boldsymbol{\tau}^i$ are computed using inverse dynamics, and the coupling forces $\boldsymbol{\lambda}^i$ are integrated as the external forces exerted by the elastic skin at the coupling points.

The second term E_S^i encourages the smoothness of the optimized motion, which is:

$$E_S^i = \|\mathbf{q}^{i+1} - 2\mathbf{q}^i + \mathbf{q}^{i-1}\|^2.$$

The two terms, E_{EE}^i and E_{COM}^i , enforce the optimized motion to follow the user-specified end-effector and COM trajectories respectively:

$$E_{EE}^i = \|\phi_{EE}^i(\mathbf{q}^i, \mathbf{u}^i) - \mathbf{e}^i\|^2, \quad E_{COM}^i = \|\phi_{COM}^i(\mathbf{q}^i, \mathbf{u}^i) - \mathbf{g}_i\|^2,$$

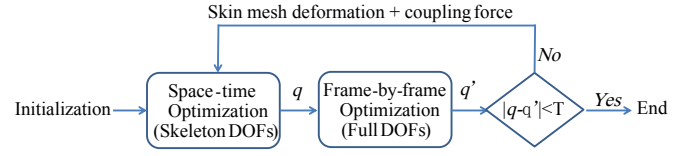


Fig. 2. Alternating algorithm Flowchart.

where the functions $\phi_{EE}^i(\mathbf{q}^i, \mathbf{u}^i)$ and $\phi_{COM}^i(\mathbf{q}^i, \mathbf{u}^i)$ define how to compute the end effector and COM positions, given the generalized coordinates of the skeleton and the deformation of the skin mesh. For each end effector, we select one vertex closest to the end effector of the rigid skeleton and represent this vertex into the local coordinate system of the end effector to compute ϕ_{EE} (see Fig. 3).

The term E_F^i penalizes the deviation from the motion $\{\tilde{\mathbf{q}}^i, i = 1, \dots, N\}$ generated in the previous frame-by-frame optimization or the initialized motion plan at the first iteration:

$$E_F^i = \|\mathbf{q}^i - \tilde{\mathbf{q}}^i\|^2.$$

After the frame-by-frame optimization step, the weight of the E_F term to follow the motion plan of the previous iteration in the space-time optimization is increased by 10%, which means the algorithm leans toward physical plausibility.

The last term E_O enforces that the end-effectors in the contact are flat:

$$E_O^i = \|\psi_{EE}^i(\mathbf{q}^i) - \hat{\mathbf{n}}\|^2,$$

where $\psi_{EE}^i(\mathbf{q}^i)$ is a function to compute the orientation of the end effector and $\hat{\mathbf{n}}$ is set to be $(0, 1, 0)$, the normal of the support plane. The formulation of this term is motivated by the fact our skinned robot is soft and hence a contact area appears whenever feet contact the ground. We try to maximize the contact area at the moment of contact because it is important to achieve a stable motion. Similarly, when the foot is about to hover, we want to clear as many contact vertices as possible. Therefore, when an end effector is close to these important moments, we add E_O to the objective function, which try to make the end effector face the upright direction of the ground.

We also impose hard kinematic and dynamic constraints on the optimized variables to obtain a stable motion plan. The kinematic constraints include the contact constraint, the center of pressure (COP) constraint and the optional periodic constraint, while the dynamic constraints include the momentum and the friction force constraints as in [36].

Contact constraint: In the optimization, we need to enforce the footfall pattern that is specified by the user to encode when the foot should leave or touch the ground. This constraint can be written into:

$$\begin{aligned} c_j^i \phi_{EE,j}^i(\mathbf{q}^i, \mathbf{u}^i)_y &= 0, & \forall i, j, \\ c_j^{i-1} c_j^i (\phi_{EE,j}^i(\mathbf{q}^i, \mathbf{u}^i) - \phi_{EE,j}^{i-1}(\mathbf{q}^i, \mathbf{u}^i)) &= 0, & \forall i, j, \end{aligned} \quad (4)$$

where c_j^i is a binary variable set to 1 if the j -th end effector is in contact with the ground at the i -th frame. This variable can be directly derived from the footfall pattern. The y coordinate of the end effector, denoted by $\phi_{EE,j}^i(\mathbf{q}^i, \mathbf{u}^i)_y$, should be 0 since the ground is set to be $y = 0$.

COP constraint: The COP should be inside the supporting polygon for a stable motion plan. This constraint can be written into:

$$\mathbf{P} \cdot \phi_{COP}^i \leq \mathbf{0}, \quad \forall i, \quad (5)$$

where the function ϕ_{COP}^i computes the COP position using the same method as in [3]. The rows in \mathbf{P} represent edges of the supporting polygon. Since the space-time optimization requires the foot to be flat on the supporting plane, the supporting polygon is formed by the convex hull of the vertices that represent the sole meshes of the end-effectors in contact. However, the contact forces and torques at a sole are simplified to be exerted on a single point of the end effector to ease the optimization. Fig. 3 illustrates the contact points (red balls) selected as the positions of the end-effector for the beetle-like robot.

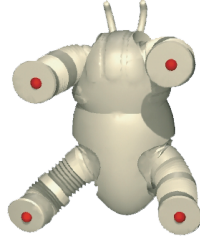


Fig. 3. Foot contacts.

Periodic constraint: When the user desires a periodic motion, the joint angles are expected to be the same in the first and the last frame of the optimized motion plan:

$$J(q^1) = J(q^N), \quad (6)$$

where J extracts the joint angles from the generalized coordinates.

Momentum constraint: The change of the linear and angular momentum of the robot should be determined by the external contact forces and torques, which can be formulated into the following equations:

$$\begin{aligned} \mathbf{R}^i &= mg + \sum_j c_j^i \mathcal{F}_{c,j}^i, & \forall i, j, \\ \mathbf{L}^i &= \sum_j c_j^i \left((\mathbf{p}_{c,j}^i - \phi_{COM}^i(\mathbf{q}^i, \mathbf{u}^i)) \times \mathcal{F}_{c,j}^i + \boldsymbol{\tau}_{c,j}^i \right), & \forall i, j, \end{aligned} \quad (7)$$

where \mathbf{R}^i and \mathbf{L}^i are, respectively, the total linear and angular momentum of the robot at i -th frame, and $\mathbf{p}_{c,j}^i$ gives the contact position of the j -th end effector.

Friction force constraint: The contact force should be inside the friction cone to satisfy the Coulomb model of friction. Specifically, we have:

$$\begin{aligned} \frac{1}{m} (\mu \mathcal{F}_{c,j,\perp}^i - \|\mathcal{F}_{c,j,\parallel}^i\|) &\geq 0, & \forall c_j^i = 1, \\ \frac{1}{m} (\|\mathcal{F}_{c,j}^i\|^2 - \|\frac{\boldsymbol{\tau}_{c,j}^i}{v_b}\|^2 - \frac{\boldsymbol{\tau}_{c,j,\perp}^i}{v_t}) &\geq 0, & \forall c_j^i = 1, \end{aligned} \quad (8)$$

where $\mathcal{F}_{c,j,\perp}^i$ and $\mathcal{F}_{c,j,\parallel}^i$ represent the components of the contact force perpendicular and parallel to the ground respectively. The first equation requires the contact force to be inside a friction cone. The second equation relates the contact force and the contact torque, where v_b and v_t are set to be the radius of the circumcircle of the sole mesh.

Optimization: With the defined objective function and constraints, the space-time optimization step can be written into:

$$\begin{aligned} \min_{\mathbf{q}^i, \mathcal{F}_{c,j}^i, \boldsymbol{\tau}_{c,j}^i, i=1..N} & E_A \\ \text{st. : } & \phi_e = 0, \phi_g \leq 0 \end{aligned} \quad (9)$$

where ϕ_e and ϕ_g represent the equality and inequality constraints respectively. We solve this sequential quadratic programming problem using the Gauss-Newton algorithm. The weights in the objective function are specified as follows: $\alpha_t = 1e-2$, $\alpha_s = 0.5$, $\alpha_c = \alpha_e = 1$, $\alpha_f = 1$ and $\alpha_o = 10$.

5.2 Frame-by-frame Optimization with Full Dynamics

In the frame-by-frame optimization, we drop the simplification of the previous step and consider the full dynamics formulated

in Eq. (30) in a similar way as in [7]. This allows us to further improve the physical plausibility of the motion. In practice, this requires the solution of a difficult quadratic programming problem with complementarity constraints (QPCC) to handle contact and friction. Different to the soft body only formulation in [7], the coupling constraints between multi-body skeleton and elastic skin introduce a large number of Lagrange multiplier variables and largely increase the complexity of the solver. To speed up the solution, we follow the condensation technique widely used in physical simulation [79], [80]. Specifically, we select the driving torques at joints and foot contact forces as optimization variables and lump the DOFs of mesh vertices and rigid skeletons to these variables through the condensation of the system matrix in Eq. (30). This choice facilitates the formulation of the physical torque limit constraints of motors as well. In this section, we describe the details of the matrix condensation and the per-frame optimization problem.

System matrix condensation: With the coupled multibody-elastic system defined in Eq.(30), the nonlinear relation between the simulated DOFs $\Delta \mathbf{u}$ and $\Delta \mathbf{q}$ and the vector \mathbf{b}_r and \mathbf{b}_d can be revealed by eliminating the unknown Lagrange multipliers $\boldsymbol{\lambda}$ in the matrix condensation (see Appendix.B for the detailed derivation):

$$\begin{aligned} \Delta \mathbf{u} &= \mathbf{A}_d^{-1} \mathbf{b}_d - \mathbf{A}_d^{-1} \nabla_u \mathcal{L}^\top \mathbf{A}_C^{-1} (\nabla_q \mathcal{L} \mathbf{A}_r^{-1} \mathbf{b}_r + \nabla_u \mathcal{L} \mathbf{A}_d^{-1} \mathbf{b}_d), \\ \Delta \mathbf{q} &= \mathbf{A}_r^{-1} \mathbf{b}_r - \mathbf{A}_r^{-1} \nabla_q \mathcal{L}^\top \mathbf{A}_C^{-1} (\nabla_q \mathcal{L} \mathbf{A}_r^{-1} \mathbf{b}_r + \nabla_u \mathcal{L} \mathbf{A}_d^{-1} \mathbf{b}_d), \end{aligned} \quad (10)$$

where $\mathbf{A}_C = \nabla_q \mathcal{L} \mathbf{A}_r^{-1} \nabla_q \mathcal{L}^\top + \nabla_u \mathcal{L} \mathbf{A}_d^{-1} \nabla_u \mathcal{L}^\top$.

According to the formulation of \mathbf{b}_r and \mathbf{b}_d in Appendix.A, the joint torques and the contact forces exerted on the soft skin are absorbed into the vector \mathbf{g}_r and \mathbf{g}_d . Since the rest terms in \mathbf{b}_r and \mathbf{b}_d are constant, we can use $\phi_{\Delta \mathbf{q}}$ and $\phi_{\Delta \mathbf{u}}$ to represent Eq. (10) more precisely:

$$\Delta \mathbf{q} = \phi_{\Delta \mathbf{q}}(\boldsymbol{\tau}, \mathcal{F}_\perp, \mathcal{F}_\parallel), \quad \text{and} \quad \Delta \mathbf{u} = \phi_{\Delta \mathbf{u}}(\boldsymbol{\tau}, \mathcal{F}_\perp, \mathcal{F}_\parallel). \quad (11)$$

Here, \mathcal{F}_\perp and \mathcal{F}_\parallel are magnitudes of normal and tangent forces at all the contact vertices on the skin mesh, and $\boldsymbol{\tau}$ represents the joint torques. To handle LCP constraints, the contact force at a contact vertex is modeled as $\mathcal{F} = \hat{\mathbf{n}} \mathcal{F}_\perp + \mathbf{D} \mathcal{F}_\parallel$ instead of the 3D vector representation in Section 5.1, where $\hat{\mathbf{n}} \mathcal{F}_\perp$ represents the upright supporting force along the contact normal $\hat{\mathbf{n}} = [0, 1, 0]^\top$ and $\mathcal{F}_\perp \in \mathbb{R}$ is the force magnitude. $\mathbf{D} \in \mathbb{R}^{3 \times 4}$ is a matrix and its columns are the vectors that span the contact plane. In our system, we use four directions to form the friction cone [81] and $\mathcal{F}_\parallel \in \mathbb{R}^4$ is the tangent magnitude. The explicit penalty forces for resolving the collision between the skeleton and skin are pre-determined quantities and merged into the constant terms in \mathbf{b}_r and \mathbf{b}_d .

Given \mathbf{q}^{i-1} and \mathbf{u}^{i-1} , $\Delta \mathbf{q}^i$ and $\Delta \mathbf{u}^i$ determine the positions and orientations of COM, COP, and end effectors at the i^{th} frame. Thanks to Eq. (11), these kinematics variables are now functions of $\boldsymbol{\tau}$, \mathcal{F}_\perp , and \mathcal{F}_\parallel . Therefore, we can derive the functions required in the computation of kinematic information, namely ϕ_{EE}^i , ψ_{EE}^i , ϕ_{COM}^i and ϕ_{COP}^i , if external forces $\boldsymbol{\tau}^i$, \mathcal{F}_\perp^i , and \mathcal{F}_\parallel^i are given. Note that the purpose of these functions are explained in the space-time optimization (see Sec. 5.1).

Optimization: We follow the control strategy used in [7] to optimize the input motion plan on a frame-by-frame basis. It is used to make sure that the output joint angle trajectories of the space-time optimization step are physically feasible, which is achieved using the two-way coupled multibody-elastic dynamics as constraints. At each frame, it can be formulated as a quadratic programming problem with complementarity constraints.

Specifically, we seek for joint torques ($\boldsymbol{\tau}^i$) and contact forces (\mathcal{F}_\perp^i , \mathcal{F}_\parallel^i) such that the corresponding $\Delta \mathbf{q}^i = \phi_{\Delta \mathbf{q}}^i$ and $\Delta \mathbf{u}^i = \phi_{\Delta \mathbf{u}}^i$

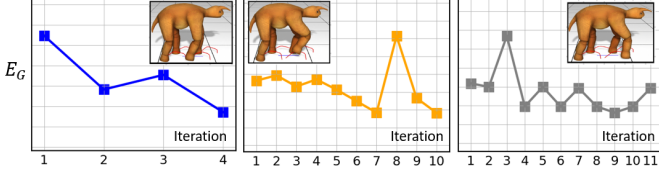


Fig. 4. Our QPCC solver converges quickly in most cases. The left plot is the converging curve of a frame when the front left leg of the monster-like robot leaves the ground. The middle plot is the converging curve of a frame when this leg is in the air (i.e. other three feet are on the ground). The right plot is the converging curve of a frame when this leg hits ground again.

satisfy necessary hard constraints and the resulting locomotion matches the input locomotion as much as possible. Mathematically, it is formulated as

$$\begin{aligned}
 & \min_{\boldsymbol{\tau}^i, \mathcal{F}_\perp^i, \mathcal{F}_\parallel^i, \boldsymbol{\lambda}_\parallel^i} E_G(\boldsymbol{\tau}^i, \mathcal{F}_\perp^i, \mathcal{F}_\parallel^i) \\
 & \text{subject to:} \\
 & \|\boldsymbol{\tau}_m^i\| < U_m, m = 1, \dots, M \\
 & \mathbf{P} \cdot \boldsymbol{\phi}_{COP}^i(\boldsymbol{\tau}^i, \mathcal{F}_\perp^i, \mathcal{F}_\parallel^i) \leq \mathbf{0} \\
 & \mathbf{0} \leq \begin{bmatrix} \mathcal{F}_\perp^i \\ \mathcal{F}_\parallel^i \\ \boldsymbol{\lambda}_\parallel^i \end{bmatrix} \perp \begin{bmatrix} \hat{\mathbf{n}}^\top \frac{\Delta \mathbf{u}_c^i}{\Delta t} \\ \mathbf{D}^\top \frac{\Delta \mathbf{u}_c^i}{\Delta t} + \mathbf{1} \boldsymbol{\lambda}_\parallel^i \\ \mu \mathcal{F}_\perp^i - \mathbf{1}^\top \mathcal{F}_\parallel^i \end{bmatrix} \geq \mathbf{0}.
 \end{aligned} \tag{12}$$

In Eq. (12), the first hard inequality constraint of $\|\boldsymbol{\tau}_m^i\| < U_m$ is the motor constraint requiring for all the M motors that the computed torque magnitude is within its physical limit U_m . The second inequality constraint $\mathbf{P} \cdot \boldsymbol{\phi}_{COP} \leq \mathbf{0}$ requires the position of the COP to be within the supporting polygon as in Sec. 5.1. The last complementary constraint is enforced at each individual contact vertex. It characterizes the contact mechanism such that when normal force exists, the relative velocity between the ground and the contact vertex along the contact normal should be zero, etc. Here, $\Delta \mathbf{u}_c^i \in \mathbb{R}^3$ is the incremental displacement of a contact vertex. The auxiliary vector $\boldsymbol{\lambda}_\parallel^i$ is related to the tangent velocity of a sliding contact; μ is the friction coefficient; and $\mathbf{1}$ is a vector of ones, that is, $\mathbf{1} = [1, 1, 1, 1]^\top$.

The objective function E_G has four terms:

$$E_G = \alpha_S E_S + \alpha_F E_F + \alpha_O E_O + \alpha_\tau E_\tau + \alpha_C E_C, \tag{13}$$

where:

$$\begin{aligned}
 E_S &= \left\| \boldsymbol{\phi}_{\Delta \mathbf{q}}^i - \boldsymbol{\phi}_{\Delta \mathbf{q}}^{i-1} \right\|^2, & E_F &= \left\| \mathbf{q}^i + \boldsymbol{\phi}_{\Delta \mathbf{q}}^i - \bar{\mathbf{q}}^{i+1} \right\|^2, \\
 E_O &= \left\| \boldsymbol{\psi}_{EE}^i - \hat{\mathbf{n}} \right\|^2, & E_\tau &= \left\| \boldsymbol{\tau}_i - \boldsymbol{\tau}_{i-1} \right\|^2 \\
 E_C &= \left\| \mathcal{E}(\boldsymbol{\phi}_{EE}^i - \boldsymbol{\phi}_{EE}^{i-1}) \right\|^2.
 \end{aligned} \tag{14}$$

The first energy term E_S is the smoothness penalty, which favors motions with consistent velocities. The term E_F penalizes the deviation from the motion $\{\bar{\mathbf{q}}^i, i = 1, \dots, N\}$ generated in the previous space-time optimization. E_O is the same soft constraint on the orientation of an end effector as in Eq. (3), which can maximize the contact area for a stable motion. E_τ is used to penalize the large variation of the control torques at joints between frames. The last term E_C imposes a penalty to moving end effectors who are responsible for supporting feet. In other words, if a foot is in contact with the ground and supporting the body, we use E_C

to reduce the risk of its possible tangent sliding. Here, \mathcal{E} is an elementary matrix that picks positions of supporting end effectors. The weighting constants for each of these penalty terms are as follows: $\alpha_S = 1$, $\alpha_F = 10$, $\alpha_O = 2$, $\alpha_\tau = 0.5$ and $\alpha_C = 10$.

5.3 Solving the QPCC

The key to solving the QPCC problem of Eq. (12) is to have a feasible configuration for all the contact vertices. Our strategy is similar to that of [7]: We flip complementary constraints when the inequality constraint reaches the boundary. Specifically, contact vertices fall into one of the three following categories:

- **Contact breakage** means that the contact vertex will leave the ground plane in the next frame, and the complementary constraints should be lifted.
- **Sliding** indicates that the contact vertex is moving within the ground plane. In this situation, the complementary constraint for its normal force \mathcal{F}_\perp becomes:

$$\mathcal{F}_\perp > 0, \quad \hat{\mathbf{n}}^\top \frac{\Delta \mathbf{u}_c^i}{\Delta t} = 0, \tag{15}$$

and the complementary constraints for the tangent force \mathcal{F}_\parallel are:

$$\begin{aligned}
 \mathcal{F}_\parallel &\geq 0, \quad \mathbf{D}^\top \frac{\Delta \mathbf{u}_c^i}{\Delta t} + \mathbf{1} \boldsymbol{\lambda}_\parallel = \mathbf{0}; \\
 \boldsymbol{\lambda}_\parallel &\geq 0, \quad \mu \mathcal{F}_\perp - \mathbf{1}^\top \boldsymbol{\lambda}_\parallel = 0.
 \end{aligned} \tag{16}$$

- **Static friction** implies that the contact vertex is fixed on the contact plane. In this case, the complementary constraint for its normal force is the same as Eq. (15). The constraints for the tangent force are:

$$\begin{aligned}
 \mathcal{F}_\parallel &\geq 0, \quad \mathbf{D}^\top \frac{\Delta \mathbf{u}_c^i}{\Delta t} + \mathbf{1} \boldsymbol{\lambda}_\parallel = \mathbf{0}; \\
 \boldsymbol{\lambda}_\parallel &= 0, \quad \mu \mathcal{F}_\perp - \mathbf{1}^\top \boldsymbol{\lambda}_\parallel \geq 0.
 \end{aligned} \tag{17}$$

The inequality constraint of $\mu \mathcal{F}_\perp - \mathbf{1}^\top \boldsymbol{\lambda}_\parallel \geq 0$ specifies the friction cone constraint in the case of static friction. d

We begin solving Eq. (12) by assuming all the contact vertices are fixed, which simplifies the original problem to

$$\begin{aligned}
 & \min_{\boldsymbol{\tau}^i, \boldsymbol{\gamma}_c} E_G(\boldsymbol{\tau}^i, \boldsymbol{\gamma}_c) \\
 & \text{subject to:} \\
 & \|\boldsymbol{\tau}_m^i\| < U_m, m = 1, \dots, M \\
 & \mathbf{P} \cdot \boldsymbol{\phi}_{COP}^i \leq \mathbf{0} \\
 & [\boldsymbol{\phi}_{\Delta \mathbf{u}}^i]_c = \mathbf{0},
 \end{aligned} \tag{18}$$

where $[\boldsymbol{\phi}_{\Delta \mathbf{u}}^i]_c$ returns the incremental displacements of all the contact vertices. This assumption of fixing all the contact vertices is realized via the Lagrange multipliers method, and the resulting multipliers $\boldsymbol{\gamma}_c$ correspond to the constraint forces at these vertices. Now, let $\boldsymbol{\gamma}_c \in \mathbb{R}^3$ be the constraint force at one of the contact vertices. It can be decomposed along normal and tangent directions as:

$$\boldsymbol{\gamma}_\perp = \hat{\mathbf{n}}^\top \boldsymbol{\gamma}_c, \quad \text{and} \quad \boldsymbol{\gamma}_\parallel = (\mathbf{I} - \hat{\mathbf{n}} \hat{\mathbf{n}}^\top) \boldsymbol{\gamma}_c. \tag{19}$$

We label all the contact vertices as contact breakage, static friction, or sliding by checking $\boldsymbol{\gamma}_\perp$ and $\boldsymbol{\gamma}_\parallel$. If $\boldsymbol{\gamma}_\perp \leq 0$, which indicates a contact breakage, we remove the constraint at the vertex in the next iteration. If $\boldsymbol{\gamma}_\perp > 0$, we further examine the magnitudes of $\mu \boldsymbol{\gamma}_\perp$ and $\|\boldsymbol{\gamma}_\parallel\|$. If $\mu \boldsymbol{\gamma}_\perp > \|\boldsymbol{\gamma}_\parallel\|$, the vertex falls into the static friction category, otherwise the vertex is considered sliding. After all the contact

vertices are labelled, we can convert the complementary constraints into a set of equality or inequality constraints, as explained in Eqs. (15), (16), and (17), and re-solve the QP optimization. It is known that QPCC is NP-complete, and few contact vertices could make the optimization procedure computationally intractable. Therefore, we simplify this procedure by grouping vertices on the planar surface of the foot mesh into five patches similar to [7]. In our experiments, we found that such initial vertex grouping often provide a good start for the QPCC solver. Typical converging curves are plotted in Fig. 4, and we stop the optimization after 10 iterations. We observe that the condensed QPCC solver is around 50x faster than the QPCC without condensation.

5.4 Initialization

Given the mechanical skeleton and the skin mesh of a robot, we first associate the mesh vertices to the links of the skeleton to obtain its skinning information. Hence, the mesh vertices can be deformed with the skeleton in the space-time optimization step, while the local coordinates of the vertices should be computed using their deformed positions and the local frame of the links at each frame. The initial motion plan are computed through the space-time optimization step without the trajectory following terms E_F^t . In this step, the skin deformation is assumed to be static and each link has additional weights from its associated vertices. Afterwards, the initial skin mesh deformation is simulated by imposing the coupling constraints in the elastic simulation of the skin, and the initial coupling force is then obtained according to the deformation of the tetrahedra connected to the coupling points [56].

6 DESIGN AND FABRICATION

Designing and fabricating a quad-robot is a challenging task. We facilitate the design by using a set of mechanical skeleton templates, and narrow the gap between professional and regular users by creating several SolidWorks scripts. This allows even an inexperienced user to tweak high-level semantic parameters. Fig. 6 shows three built-in mechanical skeleton templates provided in our system for quad-robots. Each template is built of modularized CAD parts to ease the fabrication cost. The first one is the design used in the beetle-like robot, which consists of a torso structure and four limb structures. Their exploded views are detailed in the figure as well. The torso structure has four shoulder joints that connect to its four limbs. A microcontroller board sending trigger signals to the motors is mounted inside of the torso. The limb structure includes linkage parts of an upper leg, a lower leg, and a foot. On each limb, two uniaxial motors are mounted to provide necessary rotational freedoms at the knee and the ankle. The other two templates vary in different initial poses and foot link geometries.

In the following, we describe the details of the design pipeline and the fabrication procedure respectively.

6.1 Design and Editing of Mechanical Skeleton

The design starts with a given 3D model that corresponds to the appearance of the robot. Our system extracts an initial skeletal line using the mesh contraction method [82] as shown in Fig. 5. This skeleton is actually an approximation of the medial axis of the model, and it is used as a general guide for the follow-up template embedding and editing. We employed the *modular design* idea so that the user can edit the geometry of a template mechanical component to obtain a customized mechanical skeleton

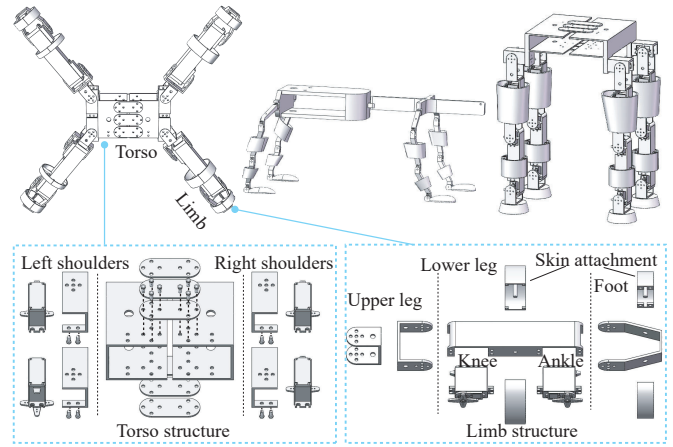


Fig. 6. Three mechanical structure templates and the exploded views of the torso and limb structures of the first template.

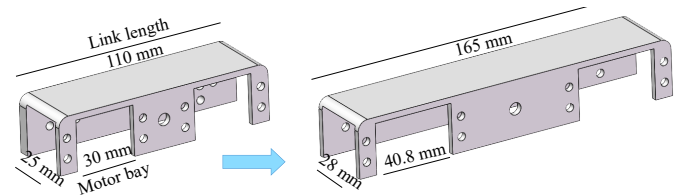


Fig. 7. With the assistance of the developed Solidworks scripts, the user only needs to tweak semantic parameters like the link length, motor mount size, etc. to obtain a customized link. The size of all the pilot holes for screw installation remains unchanged under such edits.

for quad-robots of various morphologies. To this end, several SolidWorks scripts are developed to assign semantic parameters of a link, such as the link length, motor mount size, etc., and the user only needs to tweak these intuitive parameters to obtain a personalized design without creating one from scratch. The size of pilot holes on the link for screw installation remains unchanged under such edits. An example is given in Fig. 7, where the lengths of the link and the motor bay are increased. Although a few iterations may be necessary during embedding, the developed SolidWorks scripts greatly accelerate the procedure.

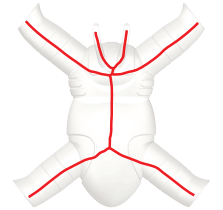


Fig. 5. The initial skeletal line of the beetle-like robot.

Typically, given a new surface model of a quad-robot, we embed limbs first and then adjust the geometry of the torso to make sure it fits the exterior skin. Specifically, a global scale of the mechanical structure template and local rotations of the links are first performed so that the template can be inside the input surface mesh. Then, the user can select the start and end points of a link on the extracted skeletal line and trigger the designed script to edit the link geometry to match the specified length and adjust the width of the link. Finally, the bracing unit of the torso is generated in a similar way as the skin creation (the details follows shortly), and we dig out holes to reduce its weight, for instance, the bracing unit for the torso of the beetle-like robot shown in the first picture in the second row of Fig. 1.

Skin and folding regions creation: The exterior skin of the robot is designed to be 8 mm thick at the foot and 4 mm thick at other parts by default, and it is created by the mesh hollow operation in

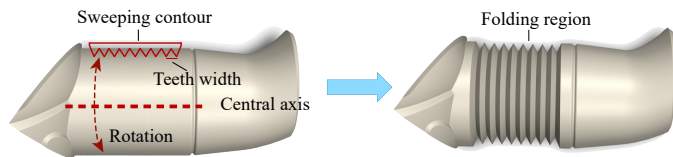


Fig. 8. We add folding regions to facilitate the stretching deformation of the skin. The template folding region is similar to gear teeth, and it is formed by a sweeping cut operation, that is, the CSG difference between the volume surrounded by the original skin surface and the volume formed by rotating the sweeping contour along a central axis.

Materialize Magics. This operation treats the space surrounded by the input 3D surface model as a solid and hollows the interior space of the solid to match the specified thickness parameters to create the skin that is amenable to fabrication. When it is being bent, the skin can yield rather large resisting forces under stretching deformations. Regular commercial motors may not possess sufficient power to overwhelm the internal stretching. To resolve this practical issue, we add a few *folding regions* on the original skin mesh, as shown in Fig. 8. The folding region is created by applying a sweeping cut operation in Solidworks over the original skin surface where the motor is installed. This small treatment increases the skin area where substantial bending occurs and effectively reduces the resulting stretching force. Note that the hollow operation is performed after the creation of the folding regions. Fortunately, the two software are compatible in mesh file format.

6.2 Fabrication

The mechanical structure of the robot is 3D printed with polypropylene-like stereolithography (SLA) resin, which is a widely used material for fabricating joints and low-friction moving parts. The exterior skin of the robot is made of a layer of soft rubbery material and fabricated via injection molding. To reduce the effort and cost of creating the skin molds, we fabricate the skin on a piece-by-piece basis: one limb has one skin piece, and the torso has two pieces as shown in Fig. 1 (Skin pieces). The skin-skeleton attachment is physically realized by another 3D printed bracing unit between the skin and the skeleton. This bracing unit is attached to each link on the skeleton and serves as a supporting structure between the rubbery skin and the mechanical skeleton (see Fig. 1 & Fig. 9). The purpose of this design is to expect that the friction between the skin and the printed parts can disable the relative motion between skin and skeleton at these parts, which is verified in the physical validation. We thus select the glue vertices in Fig. 9 according to the position of skin-skeleton attachment parts so that the coupling constraints can reflect this physical setting. The mass matrix and the inertia tensor of this bracing unit are integrated in our multibody subsystem dynamics. Finally, skin pieces are glued together after all skin pieces are installed using nonreactive PVA adhesive.

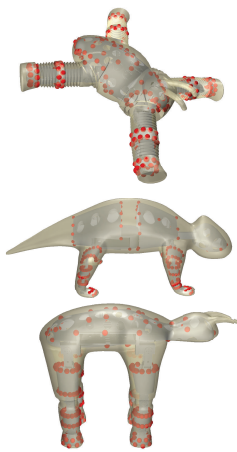


Fig. 9. Glue vertices.

Motor specification We use the MG995R servo motor to drive the motion of the skinned robot. The motor's size is $40.8 \times 20 \times 38$ mm

Robot	Joints	Skin	Glue	Weight	Opt.
Beetle-like	12	14,152	140	3.9kg	~ 3.2 s
Monster-like	16	16,624	301	11.87kg	~ 8.98 s
Lizard-like	17	18,261	176	10.8kg	~ 8.41 s

TABLE 1

Physical and simulation statistics of three tested robots. Joints: the number of joints on the skeleton. Skin: the number of vertices on the skin mesh. Glue: the number of glue vertices. Weight: the physical weight of the robot. Opt.: the average time used for optimizing Eq. (12) of one motion frame.

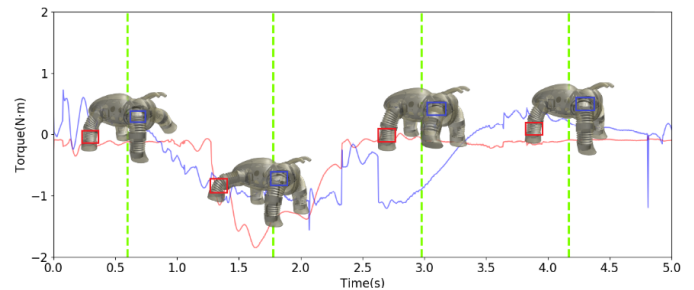


Fig. 10. We plot the torque values of the optimized motion planning of the beetle-like robot. Orange curve: the torque curve at an ankle joint (at the orange box). Blue curve: the torque curve at a shoulder joint (at the orange box).

with the maximum torque of 20 kg·cm under 6.4 V (i.e. $U_m = 1.96$ N·m in Eq. (12)). In total, 12 motors are installed in the beetle-like robot. All the motors are controlled with an Arduino board, which supports up to 32 motors.

7 EXPERIMENTAL RESULTS

In this section, we first report the torque limits and folding region experiments in the motion design of the beetle-like robot to validate its physical feasibility. The mechanical skeleton of this robot is designed based on the first template in Fig. 6 and fabricated using 3D printing. A comparison to kinematic optimization only is also provided for this robot. Second, we report the motion design results for two additional quad-robots: a monster-like robot and a lizard-like one. The performance of our optimization algorithm depends on the number of vertices on the skin mesh, the number of glue vertices, and the number of joints of the mechanical skeleton. The frame interval Δt is 0.005 second, and we employ the discrete collision detection algorithm to handle self-collisions and foot-ground collisions. Normally a motion cycle has around 500 frames. Our optimization algorithm was implemented on a desktop PC with an intel i7-7700 CPU and 16 GB memory. The soft skin is made of isotropic rubber material whose Young's modulus is 0.09 GPa, and Poisson's ratio is 0.46. Table 1 reports some essential physical and simulation statistics of these three examples. The space-time optimization step with approximated skin deformation for a skeleton is around 40 seconds. For the slow-walking motions as shown in Fig. 1 and Fig. 19, we only need one iteration to converge, since the frame-by-frame optimization can reproduce the motion from space-time optimization step well with physical constraints. For the relatively fast trotting motion in Fig. 18, the algorithm converges after two iterations.

Torque limit: The torque limit in the motion optimization (i.e. Eq. (12)) of the beetle-like robot is set as 1.96 N·m to match the physical torque limit of the used MG995R motor. To verify if this hard constraint is faithfully enforced during the optimization, we examine torque values that are calculated by our motion design

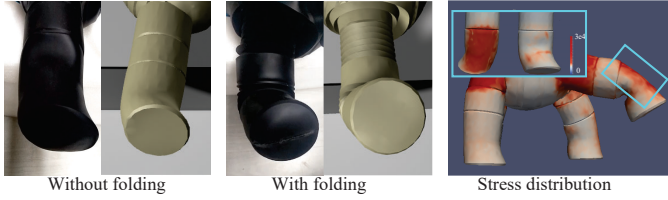


Fig. 11. Adding folding regions significantly relieves the stretching stress over the skin. We program the motor at an ankle joint of the beetle-like robot to rotate $\pm 70^\circ$ within 2 seconds in this experiment. A smooth skin can only be bent around $\pm 20^\circ$, while the folded skin is able to reach the desired deformation. The physical experiment (black skins) results are consistent with simulation results (yellow skins).

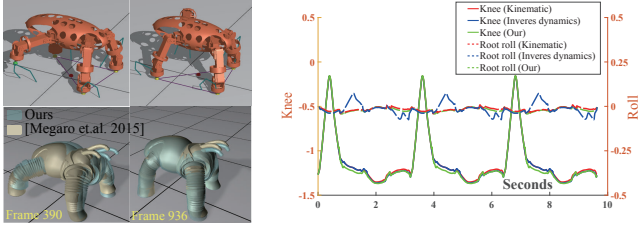


Fig. 12. The motion plan generated by the kinematic-only optimization [3] leads to unstable walking sequences. Left: Selected frames. Right: The joint angle curves. Kinematic: the kinematic optimization result. Inverse dynamics: the simulation result by following the kinematic optimization. Our: our optimization result. The large roll angles of the root link is due to the unstable pose using kinematic-only optimization. Please see the accompanying video for the full comparison.

system after per-frame optimization. The result is reported in Fig. 10, where torque curves at an ankle joint and a shoulder joint are plotted. It can be seen that the imposed motor constraint successfully bounds the torque magnitude to be within the limit to ensure that the designed motion is physically possible.

Folding regions Adding folding regions to the robot’s skin is an effective fabrication artistry to enable the robot assembly using off-the-shelf servo motors and lower the fabrication cost. To demonstrate its advantage, we compare the skin deformation under the joint rotation when the robot is attached to a regular soft skin and a folded skin. Both skins are fabricated using the same materials. It can be seen from Fig. 11 that rotating joints yield large stretching stress over the skin, which could easily go beyond the physical capacity of many commercial servo motors. In this test, we follow the aforementioned motor specification by setting the maximal torque to 1.96 N·m and test if this power is sufficient to generate the necessary skin deformation. We set our target bending angle to $\pm 70^\circ$, which is a common value in many walking gaits. The motor is programmed to reach this target in 2 seconds. Our simulation shows that the smooth robot skin without folding region prevents the motor from producing sufficient joint rotation and the maximum angle that can be reached is only about $\pm 20^\circ$. With the folding region, swept by an 8 mm-depth tooth over the smooth skin, our simulation predicts that the motor is able to generate the desired rotation. The physical experiment results are quite consistent with our simulation prediction as reported side by side in Fig. 11.

VS. kinematic-only optimization: In contrast to robots with only rigid mechanical skeletons, skinned robots exhibit a much more complicated dynamic behavior, which should be fully incorporated during the motion design. To illustrate the necessity

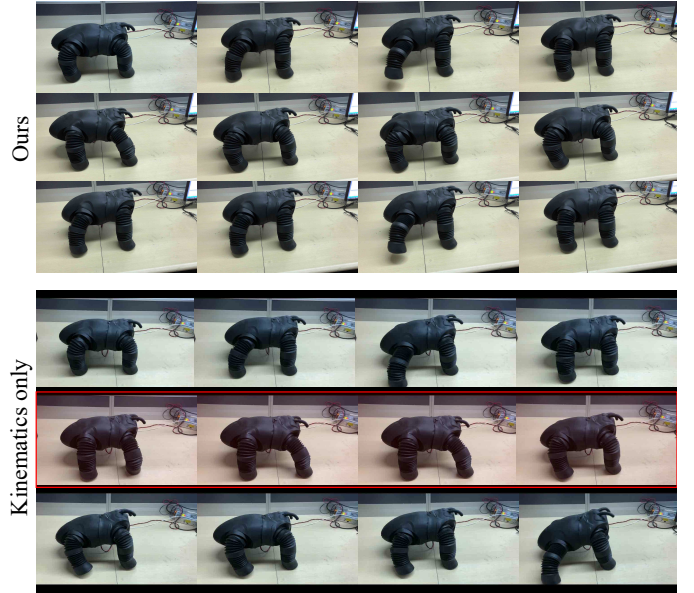


Fig. 13. Physical experiments show that kinematic-only optimization is not a feasible solution for skinned robots – there are noticeable back steps (highlighted with a red box) in a motion cycle as the driving torques, after damped by the skin deformation, are not strong enough to produce necessary normal contact forces. Please refer to the accompanied video for a clearer comparison.

of incorporating influences of the soft skin, we compare the motion plans generated using our method and the one by Megaro et al. [3]. Because the primary focus in [3] is to design robot creatures with only rigid links, Megaro and colleagues used a kinematic-based optimization strategy, which includes the trajectories of COM, COP, end effectors, and the footfall pattern. Based on the resulting motion plan, we compute the corresponding driving torques at joints using inverse dynamics. Specifically, the driving torques are computed by imposing another set of rotation constraints over the skeleton in Eq. (30) using the Lagrange multipliers method (with necessary complementary constraints and inequality constraints to handle the ground contact and motor torque limit). The constrained joint rotation corresponds to the one obtained from the kinematic-based motion plan, and the multipliers represent required generalized constraint force, which are converted to joint torque via \mathbf{J}_{ok} to achieve the target joint rotation. As shown in Fig. 12, the physical simulation results suggest that a kinematically valid joint trajectory does not guarantee a smooth walking cycle of the skinned robot, even though the constraints of COM/COP are also specified in the kinematic optimization without skin information. The coincidence, for the knee joint, of the joint angle curves of kinematic and inverse dynamics shows that our inverse dynamics computation can track the kinematic motion plan well, and the roll angle of the root link experiences a larger variation in the inverse dynamics simulation. This is also verified in the physical experiment as shown in Fig. 13. The motion plan obtained using only kinematic optimization leads to a shaky motion. We also observe backward motions as highlighted in the figure. Our method, because it fully considers various physics conditions and constraints, yields a much better result.

Motion design results: Fig. 14 illustrates the simple foot lifting motions designed by our system for the beetle-like robot. These two motions, i.e., single-foot lifting and double-foot lifting, are

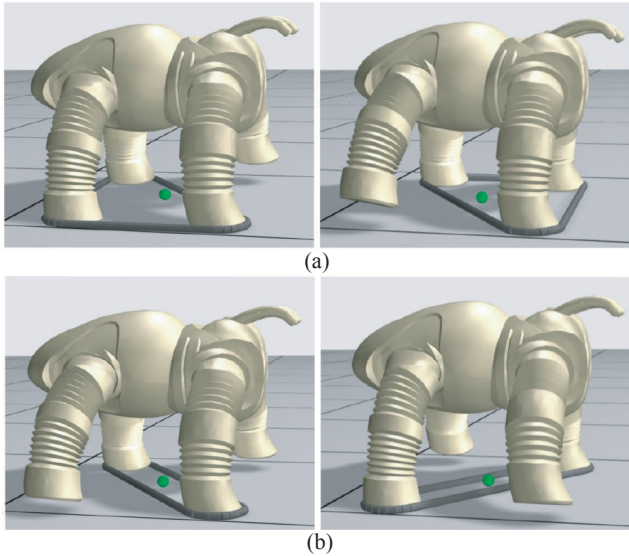


Fig. 14. The foot lifting motion for the Beetle-like robot. (a) Single-foot lifting. (b) double-foot lifting. The green balls indicate the COP positions and gray lines the support polygons. Our optimization algorithm can constrain the COP to be inside the support polygon.

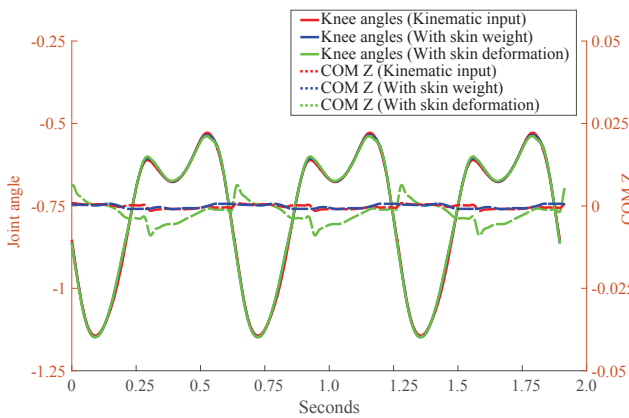


Fig. 15. The comparison of joint angle curves and COM positions. With skin weight: the curves are from the initial space-time optimization result with only skin weight. With skin deformation: curves from the space-time optimization result with skin deformation simulated by frame-by-frame optimization. COM Z: the z component of the COM, representing the COM movement from left to right during the motion.

also used to show the COP is constrained to be inside the support polygon with our optimization algorithm. Please see the accompanying video for the animation.

The embedded skeleton of the monster-like robot shown in Fig. 6 is designed using the third template. The weight of this robot is 11.87 kg, and its size is 48.5 cm × 64.6 cm × 27 cm. The young modulus at the tail and belly of the monster is reduced by 85% to demonstrate the dynamics of the skin. Two different input foot trajectories are used to generate the walking motions for the robot. As shown in the leftmost column in Fig. 17, the first trajectory has a longer stride length but lower step, while the second one has shorter stride and higher step. Our system is able to accommodate such variations and produces a smooth and physically correct motion plan. The walking speed for these two walking motions are 0.11 meter/second and 0.06 meter/second.

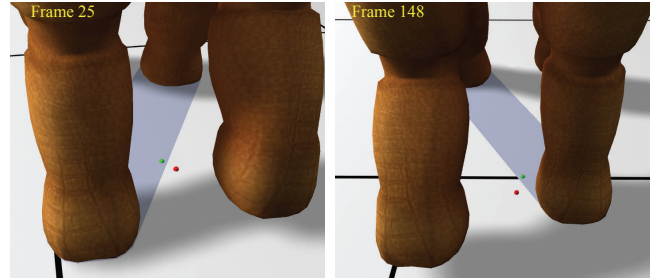


Fig. 16. The effect of the COP constraints in the trotting motion plan. Red balls: The COP positions computed using the motion plan after initialization. Green balls: The COP positions optimized with the skin mesh deformation. The supporting polygons are in cyan.

We generate a trotting motion of 0.45 meter/second speed for the monster-like robot to demonstrate that our system can support fast motion. In this example, the trajectories of joint positions are labelled using the horse motion pictures photographed by Eadweard Muybridge, a famous photographer for his work on motions. The joint positions are mapped to a horse motion with the specified speed using the method in [83] and re-targeted to the skeleton of our monster. This initial kinematic motion (please see the accompanying video for the motion) is then optimized using our alternating motion optimization algorithm to turn it into a physically feasible motion.

We notice that the fighting phase of the initial kinematic motion is not consistent with the foot contact plan. To be more specific, the time of the fighting phase is not enough for the monster to return back to the ground. Thus, the space-time optimization with physical constraints, especially the momentum constraint, is necessary to eliminate such inconsistency, which is critical to the success of the alternating optimization. Fig. 16 illustrates the effect of the COP optimization. The red balls indicate that the COP positions at two frames in the initialized trotting motion plan are outside of supporting polygon after the first space-time optimization that does not account for the deformation of the skin mesh. Thus, the frame-by-frame simulation fails to produce a stable trotting motion with its initially optimized motion plan, as shown in the second row of Fig. 18. The COP constraint is turned off to produce this failed example once this constraint can not be satisfied by the solver. After the second iteration, the COPs are moved into supporting polygons, as indicated by the green balls. A smooth trotting motion can then be generated as shown in the first row on the right of Fig. 18. The comparisons of the optimized joint angles and the z components of COP are illustrated in Fig. 15. The variation of z components indicates the COP moves from left to right so that it is inside the supporting region.

Another example is reported in Fig. 19. The mechanical skeleton of this robot is further edited based on the second template of Fig. 6. We lengthened the torso and shortened the limbs to fit this template into the input model (125 cm × 47 cm × 46 cm). Our system also produces plausible motion plans for this quad-robot.

Failure case: While our system is stable in the generation of slow walking motion, we find the generation of fast trot motion is sensitive to the physical parameters. When the mass of the monster is increased to two times, its influence on the mass center cannot be balanced in the optimization and the COP constraint is violated in the space-time optimization result, possibly due to its conflict between the foot contact constraint. Hence, the frame-by-frame

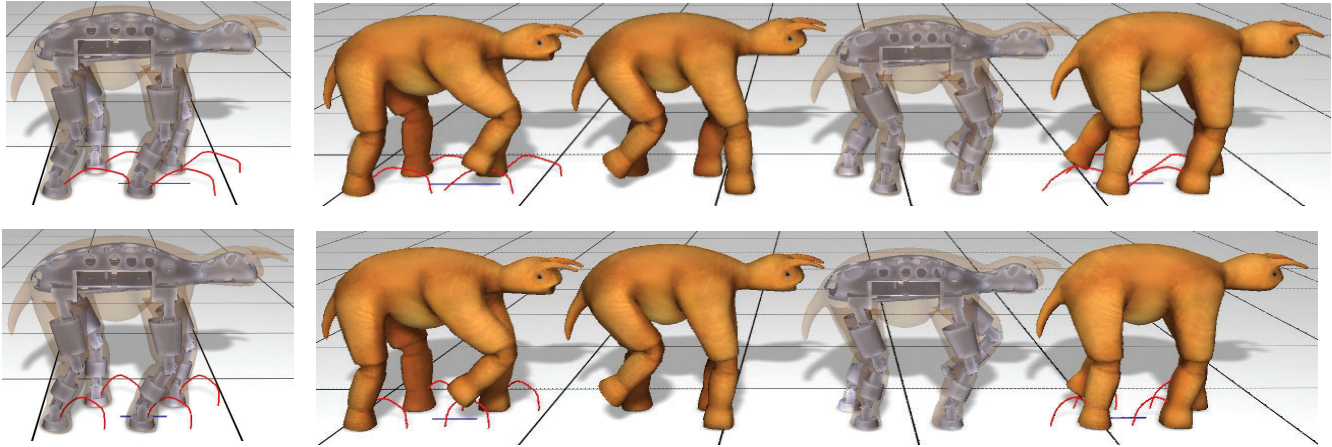


Fig. 17. The monster-like robot takes two different input foot trajectories, and our system computes natural and physically correct motions for both inputs.

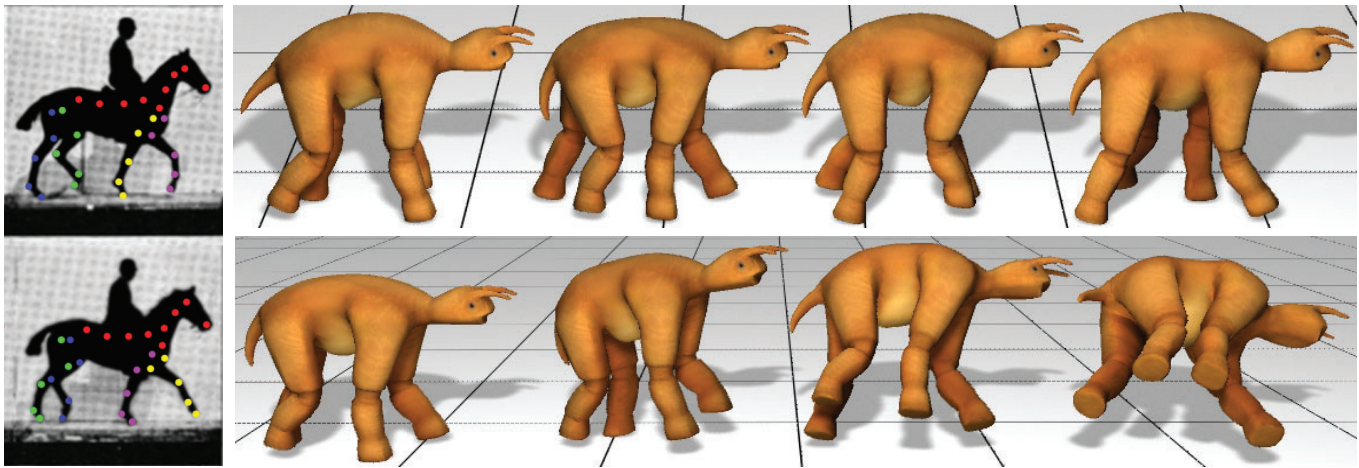


Fig. 18. Trotting motion for the monster-like robot. (a) Marked joint positions. The color of a dot indicates to which part of the horse skeleton it belongs, and the positions are mapped to our monster skeleton joint angles using the space-time optimization with only kinematic constraints. (b) The designed trotting motion with our alternating algorithm. (c) The unstable motion simulated by the frame-by-frame optimization when the skin deformation is not considered in the space-time optimization.

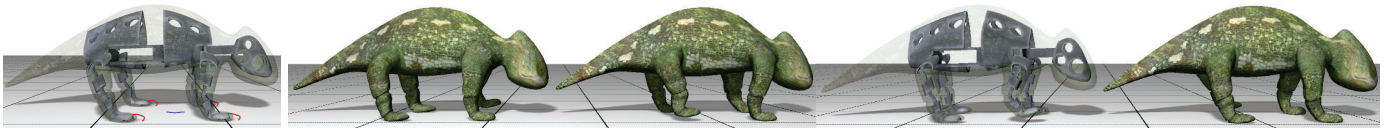


Fig. 19. The motion of a lizard-like robot. We edit the second template in Fig. 6 with SolidWorks scripts to create the design of its mechanical skeleton. With user provided inputs, our system generates plausible motions of this robot.

optimization will fail to produce a stable motion. Such situation might be handled through the integration of foot plan sampling step in [36].

8 CONCLUSION AND FUTURE WORK

In this paper, we have presented a fabrication-oriented motion planing algorithm and detailed design/fabrication procedures for personalized skinned quad-robots. The physical constraints, such as the equations of motion of the skinned robot and the motor constraints, are integrated into the motion planning such that the resulting motion plan is physically and dynamically feasible.

The condensation formulation allows us to conveniently establish the nonlinear relationship between external forces and the target kinematic parameters of the locomotion and to reach a QPCC formulation for the motion design. Our experiments show that the system is able to assist regular users to obtain natural and smooth motions designed for skinned quad-robots.

In the future, we want to explore a gait synthesis algorithm to generate the motion plan from high-level parameters, such as velocity and turning angles. Combining captured gait data and optimization with dynamic constraints has the potential to significantly reduce users' labor efforts of creating such motion planning.

Currently, the coupling between the FEM simulation of the soft skin and the rigid body dynamics of the mechanical structure is not fast enough for a closed-loop control of skinned quad-robots. We want to explore model reduction or homogenization techniques to reduce the computational cost of FEM simulation and produce interactive feedback to control skinned robots online.

APPENDIX A THE EQUATIONS OF MOTION

Lagrangian multibody dynamics: The multibody rigid skeleton of the robot is a kinematic tree of links connected by joints. Its classic Lagrangian mechanics formulation can be found in [84], where the DOFs of the skeleton are specified as the the generalized coordinate \mathbf{q} of the joint angles and the root translation.

The equation of motion for the articulated rigid skeleton can be written as

$$\mathbf{M}_r(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{D}_r\dot{\mathbf{q}} + \mathbf{f}_r(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}_r, \quad (20)$$

where the subscript $[\cdot]_r$ denotes variables for the skeleton rig. Detailed derivation of \mathbf{M}_r and \mathbf{f}_r in Eq. (20) can be found in the excellent tutorial by Liu and Jain [85]. \mathbf{D}_r is the damping matrix. We refer to \mathbf{M}_r as the *rigid mass matrix* in order to differentiate it from the mass matrix of the soft skin. For a skeleton with K links,

$$\mathbf{M}_r = \sum_{k=1}^K \mathbf{J}_k^\top \mathbf{M}_{ck} \mathbf{J}_k, \quad \text{and} \quad \mathbf{M}_{ck} = \begin{bmatrix} m_k \cdot \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{ck} \end{bmatrix}, \quad (21)$$

where \mathbf{I} is the identity matrix, \mathbf{I}_{ck} is the inertia tensor for the k^{th} rigid link, and m_k is the mass of the link. The Jacobian matrix $\mathbf{J}_k = [\mathbf{J}_{vk}^\top, \mathbf{J}_{\omega k}^\top]^\top$, where $\mathbf{J}_{vk} = \partial \mathbf{x}_k / \partial \mathbf{q}$, relates the Cartesian coordinate \mathbf{x}_k of the link's COM to the generalized coordinate \mathbf{q} . Similarly, $\mathbf{J}_{\omega k}$ relates the angular velocity $\boldsymbol{\omega}_k$ to the generalized velocity $\dot{\mathbf{q}}$ such that $\boldsymbol{\omega}_k = \mathbf{J}_{\omega k} \dot{\mathbf{q}}$. It is noteworthy that the rigid mass matrix is not constant because of the orientation-dependent inertia tensor and the Jacobian matrix.

Non-inertia forces like Coriolis and centrifugal forces that couple the generalized coordinate are included in \mathbf{f}_r . The right-hand side of Eq. (20) is the generalized external force applied to the skeleton, which includes the gravity force \mathbf{g} and torques $\boldsymbol{\tau}$ from the actuating motors:

$$\mathbf{g}_r = \sum_{k=1}^K \mathbf{J}_k^\top \begin{bmatrix} \mathbf{g}_k \\ \boldsymbol{\tau}_k \end{bmatrix} = \sum_{k=1}^K [\mathbf{J}_{vk}^\top, \mathbf{J}_{\omega k}^\top] \begin{bmatrix} \mathbf{g}_k \\ \boldsymbol{\tau}_k \end{bmatrix}. \quad (22)$$

FEM elastic simulation: The dynamics of the soft skin can also be formulated using Lagrangian mechanics, and we can obtain the equation of motion in a similar form:

$$\mathbf{M}_d \ddot{\mathbf{u}} + \mathbf{D}_d \dot{\mathbf{u}} + \mathbf{f}_d(\mathbf{u}) = \mathbf{g}_d. \quad (23)$$

The subscript $[\cdot]_d$ denotes variables for the deformable skin, and \mathbf{g}_d is the external force applied to the soft skin. We discretize the volume of the skin by a tetrahedral mesh. The *deformable mass matrix* \mathbf{M}_d is constant and can be assembled using the standard FEM [86]. \mathbf{D}_d is the damping matrix, and \mathbf{f}_d denotes the internal elastic force, and it is the negative gradient of the strain energy Ψ such that $\mathbf{f}_d = -\nabla \Psi$. The specific formulation of Ψ depends on the material model chosen. Typically, it is computed based on three *isotropic invariants* of the deformation gradient tensor \mathbf{F} :

$$I_1(\mathbf{F}) = \text{tr}(\mathbf{F}^\top \mathbf{F}), \quad I_2(\mathbf{F}) = \text{tr}\left(\frac{1}{2}(\mathbf{F}^\top \mathbf{F})^2\right), \quad I_3(\mathbf{F}) = \det(\mathbf{F}^\top \mathbf{F}). \quad (24)$$

For the robot with soft skin, its articulated skeleton motion will induce large local compression, especially near the joint. Material models like the StVK and co-rotational models that have been widely used in previous research [72], [74] become unstable under extreme compression. We therefore opted to use the Neo-Hookean material, whose strain energy density is defined as

$$\Psi \triangleq \frac{\mu}{2}(I_1 - \log(I_3) - 3) + \frac{\lambda}{8} \log^2(I_3), \quad (25)$$

where μ and λ are Lamé constants. The first Piola-Kirchhoff stress tensor $\mathbf{P} \in \mathbb{R}^{3 \times 3}$ can be computed based on Eq. (25) using the chain rule:

$$\mathbf{P} = \frac{\partial \Psi}{\partial \mathbf{F}} = \frac{\partial \Psi}{\partial I_1} \cdot \frac{\partial I_1}{\partial \mathbf{F}} + \frac{\partial \Psi}{\partial I_3} \cdot \frac{\partial I_3}{\partial \mathbf{F}} = \mu \mathbf{F} - \mu \mathbf{F}^{-\top} + \frac{\lambda \log(I_3)}{2} \mathbf{F}^{-\top}. \quad (26)$$

The final formulation of \mathbf{f}_d is

$$\mathbf{f}_d = - \int \frac{\partial \Psi}{\partial \mathbf{u}} dV = - \int \frac{\partial \Psi}{\partial \mathbf{F}} : \frac{\partial \mathbf{F}}{\partial \mathbf{u}} dV = - \int \left(\mathbf{P} : \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right)^\top dV. \quad (27)$$

Here, $\partial \mathbf{F} / \partial \mathbf{u} \in \mathbb{R}^{3 \times 3 \times 3}$ is a third-order tensor. For a tetrahedral element with linear shape functions, $\partial \mathbf{F} / \partial \mathbf{u}$ is constant and can be precomputed and stored at each element.

Implicit backward euler time integration: We discretize the equation of motions for both the rigid skeleton and deformable body using the implicit backward Euler method to improve the stability of the simulation. Let $\Delta \mathbf{q} = \mathbf{q}^{i+1} - \mathbf{q}^i$ and $\Delta \dot{\mathbf{q}} = \dot{\mathbf{q}}^{i+1} - \dot{\mathbf{q}}^i$, where the superscript $[\cdot]^i$ is the frame index. Given the time interval Δt between frame i and $i+1$, the velocity and acceleration of \mathbf{q} at frame $i+1$ can be discretized as: $\dot{\mathbf{q}}^{i+1} = \Delta \mathbf{q} / \Delta t$ and $\ddot{\mathbf{q}}^{i+1} = \Delta \dot{\mathbf{q}} / \Delta t = \Delta \mathbf{q} / \Delta t^2 - \dot{\mathbf{q}}^i / \Delta t$. The velocity and acceleration of \mathbf{u} can be derived similarly. Subsequently, Eqs. (20) and (23) can be linearized as

$$(\mathbf{M}_r(\mathbf{q}^i) + \Delta t \mathbf{D}_r + \mathbf{C}(\mathbf{q}^i, \dot{\mathbf{q}}^i)) \Delta \mathbf{q} = \Delta t^2 (\mathbf{g}_r - \mathbf{D}_r \dot{\mathbf{q}}^i), \quad (28)$$

and

$$\left(\mathbf{M}_d + \Delta t \mathbf{D}_d + \Delta t^2 \frac{\partial \mathbf{f}_d}{\partial \mathbf{u}^i} \right) \Delta \mathbf{u} = \Delta t^2 (\mathbf{g}_d - \mathbf{f}_d(\mathbf{u}^i) - \mathbf{D}_d \dot{\mathbf{u}}^i). \quad (29)$$

In Eq. (28), we compute $\mathbf{M}_r(\mathbf{q}^i)$ and $\mathbf{C}(\mathbf{q}^i, \dot{\mathbf{q}}^i)$ using state variables at frame i . Therefore, Eq. (20) is only semi-implicitly discretized [87], [88].

Linear systems: The two-way coupled multibody-elastic system in Eq.[2] in Sec. 4.1 of our paper is as follows:

$$\begin{bmatrix} \mathbf{A}_r & \mathbf{0} & \nabla_q \mathcal{L}^\top \\ \mathbf{0} & \mathbf{A}_d & \nabla_u \mathcal{L}^\top \\ \nabla_q \mathcal{L} & \nabla_u \mathcal{L} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{q} \\ \Delta \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_r \\ \mathbf{b}_d \\ \mathbf{0} \end{bmatrix}. \quad (30)$$

The matrices $\mathbf{A}_r, \mathbf{A}_d$ in this equation are just the abbreviation of system matrices in Eqs. (28) and (29), where $\mathbf{A}_r = \mathbf{M}_r(\mathbf{q}^i) + \Delta t \mathbf{D}_r + \Delta t^2 \mathbf{C}(\mathbf{q}^i, \dot{\mathbf{q}}^i)$ and $\mathbf{A}_d = \mathbf{M}_d + \Delta t \mathbf{D}_d + \Delta t^2 \partial \mathbf{f}_d / \partial \mathbf{u}^i$. Analogically, we have $\mathbf{b}_r = \Delta t^2 (\mathbf{g}_r - \mathbf{D}_r \dot{\mathbf{q}}^i - \mathbf{C}(\mathbf{q}^i, \dot{\mathbf{q}}^i) \dot{\mathbf{q}}^i)$, and $\mathbf{b}_d = \Delta t^2 (\mathbf{g}_d - \mathbf{f}_d(\mathbf{u}^i) - \mathbf{D}_d \dot{\mathbf{u}}^i)$.

APPENDIX B SYSTEM MATRIX CONDENSATION

The system matrix condensation in Sec.5.2 starts with eliminating the Lagrange multipliers $\boldsymbol{\lambda}$ in Eq. (30) (the same equation with Eq.[2] in Sec. 4.1). We first expand the second line of this equation, which yields:

$$\mathbf{A}_d \Delta \mathbf{u} + \nabla_u \mathcal{L}^\top \boldsymbol{\lambda} = \mathbf{b}_d, \quad \text{or} \quad \Delta \mathbf{u} = \mathbf{A}_d^{-1} (\mathbf{b}_d - \nabla_u \mathcal{L}^\top \boldsymbol{\lambda}). \quad (31)$$

Similarly, expanding the first line in Eq. (30), we can produce an equation similar to Eq. (31) but for skeleton DOFs:

$$\Delta \mathbf{q} = \mathbf{A}_r^{-1} \left(\mathbf{b}_r - \nabla_q \mathcal{L}^\top \boldsymbol{\lambda} \right). \quad (32)$$

Expanding the third line of Eq. (30) gives the linearized position constraint:

$$\nabla_q \mathcal{L} \Delta \mathbf{q} + \nabla_u \mathcal{L} \Delta \mathbf{u} = \mathbf{0}. \quad (33)$$

Substituting both Eqs. (31) and (32) into Eq. (33) yields:

$$\boldsymbol{\lambda} = \mathbf{A}_C^{-1} \left(\nabla_q \mathcal{L} \mathbf{A}_r^{-1} \mathbf{b}_r + \nabla_u \mathcal{L} \mathbf{A}_d^{-1} \mathbf{b}_d \right), \quad (34)$$

where $\mathbf{A}_C = \nabla_q \mathcal{L} \mathbf{A}_r^{-1} \nabla_q \mathcal{L}^\top + \nabla_u \mathcal{L} \mathbf{A}_d^{-1} \nabla_u \mathcal{L}^\top$. By substituting Eq. (34) back into Eqs. (31) and (32), we obtain the condensed formulas in Eq.[10] in our paper.

REFERENCES

- [1] C. Leger, “Automated synthesis and optimisation of robot configurations: An evolutionary approach,” Ph.D. dissertation, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA, 1999, cMU-RI-TR-99-43.
- [2] R. Desai, Y. Yuan, and S. Coros, “Computational abstractions for interactive design of robotic devices,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1196–1203.
- [3] V. Megaro, B. Thomaszewski, M. Nitti, O. Hilliges, M. Gross, and S. Coros, “Interactive design of 3d-printable robotic creatures,” *ACM Trans. Graph.*, vol. 34, no. 6, pp. 216:1–216:9, Oct. 2015.
- [4] P. Gallagher and M. Maclachlan, “Adjustment to an artificial limb: a qualitative perspective,” *Journal of health psychology*, vol. 6, no. 1, pp. 85–100, 2001.
- [5] C. Majidi, “Soft robotics: a perspective - current trends and prospects for the future,” *Soft Robotics*, vol. 1, no. 1, pp. 5–11, 2014.
- [6] Z. Wang, G. Hang, J. Li, Y. Wang, and K. Xiao, “A micro-robot fish with embedded sma wire actuated flexible biomimetic fin,” *Sensors and Actuators A: Physical*, vol. 144, no. 2, pp. 354–360, 2008.
- [7] J. Tan, G. Turk, and C. K. Liu, “Soft body locomotion,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 26:1–26:11, Jul. 2012.
- [8] X. Chen, H. Li, C.-W. Fu, H. Zhang, D. Cohen-Or, and B. Chen, “3d fabrication with universal building blocks and pyramidal shells,” in *SIGGRAPH Asia 2018 Technical Papers*, ser. SIGGRAPH Asia ’18, 2018, pp. 189:1–189:15.
- [9] C. Dai, C. C. L. Wang, C. Wu, S. Lefebvre, G. Fang, and Y.-J. Liu, “Support-free volume printing by multi-axis motion,” *ACM Trans. Graph.*, vol. 37, no. 4, pp. 134:1–134:14, Jul. 2018.
- [10] Y. Lan, Y. Dong, F. Pellacini, and X. Tong, “Bi-scale appearance fabrication,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 145:1–145:12, Jul. 2013.
- [11] D. Chen, D. I. W. Levin, P. Didyk, P. Sithi-Amorn, and W. Matusik, “Spec2fab: A reducer-tuner model for translating specifications to 3d prints,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 135:1–135:10, Jul. 2013.
- [12] A. Brunton, C. A. A. Arikan, T. M. Tanksale, and P. Urban, “3d printing spatially varying color and translucency,” *ACM Trans. Graph.*, vol. 37, no. 4, pp. 157:1–157:13, Jul. 2018.
- [13] K. Sakurai, Y. Dobashi, K. Iwasaki, and T. Nishita, “Fabricating reflectors for displaying multiple images,” *ACM Trans. Graph.*, vol. 37, no. 4, pp. 158:1–158:10, Jul. 2018.
- [14] B. Bickel, M. Bäcker, M. A. Otaduy, H. R. Lee, H. Pfister, M. Gross, and W. Matusik, “Design and fabrication of materials with desired deformation behavior,” *ACM Trans. Graph.*, vol. 29, no. 4, pp. 63:1–63:10, Jul. 2010.
- [15] M. Skouras, B. Thomaszewski, S. Coros, B. Bickel, and M. Gross, “Computational design of actuated deformable characters,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 82:1–82:10, Jul. 2013.
- [16] J. Panetta, Q. Zhou, L. Malomo, N. Pietroni, P. Cignoni, and D. Zorin, “Elastic textures for additive fabrication,” *ACM Trans. Graph.*, vol. 34, no. 4, pp. 135:1–135:12, Jul. 2015.
- [17] D. Chen, D. I. W. Levin, W. Matusik, and D. M. Kaufman, “Dynamics-aware numerical coarsening for fabrication design,” *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–15, Jul 2017.
- [18] M. Lau, A. Ohgawara, J. Mitani, and T. Igarashi, “Converting 3d furniture models to fabricatable parts and connectors,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 85:1–85:6, Jul. 2011.
- [19] J. Cali, D. A. Calian, C. Amati, R. Kleinberger, A. Steed, J. Kautz, and T. Weyrich, “3d-printing of non-assembly, articulated models,” *ACM Trans. Graph.*, vol. 31, no. 6, pp. 130:1–130:8, Nov. 2012.
- [20] M. Bäcker, B. Bickel, D. L. James, and H. Pfister, “Fabricating articulated characters from skinned meshes,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 47:1–47:9, Jul. 2012.
- [21] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel, “Computational design of mechanical characters,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 83:1–83:12, Jul. 2013.
- [22] D. Ceylan, W. Li, N. J. Mitra, M. Agrawala, and M. Pauly, “Designing and fabricating mechanical automata from mocap sequences,” *ACM Trans. Graph.*, vol. 32, no. 6, pp. 186:1–186:11, Nov. 2013.
- [23] R. Zhang, T. Auzinger, D. Ceylan, W. Li, and B. Bickel, “Functionality-aware retargeting of mechanisms to 3d shapes,” *ACM Trans. Graph.*, vol. 36, no. 4, pp. 81:1–81:13, Jul. 2017.
- [24] V. Megaro, J. Zehnder, M. Bcher, S. Coros, M. Gross, and B. Thomaszewski, “A computational design tool for compliant mechanisms,” *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–12, Jul 2017.
- [25] M. Geilinger, R. Poranne, R. Desai, B. Thomaszewski, and S. Coros, “Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels,” *ACM Trans. Graph.*, vol. 37, no. 4, pp. 160:1–160:12, Jul. 2018.
- [26] B. Bickel, P. Kaufmann, M. Skouras, B. Thomaszewski, D. Bradley, T. Beeler, P. Jackson, S. Marschner, W. Matusik, and M. Gross, “Physical face cloning,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 118:1–118:10, Jul. 2012.
- [27] J. M. Bern, K.-H. Chang, and S. Coros, “Interactive design of animated plushies,” *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–11, Jul 2017.
- [28] L.-K. Ma, Y. Zhang, Y. Liu, K. Zhou, and X. Tong, “Computational design and fabrication of soft pneumatic objects with desired deformations,” *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–12, Nov 2017.
- [29] A. Witkin and M. Kass, “Spacetime constraints,” *SIGGRAPH Comput. Graph.*, vol. 22, no. 4, pp. 159–168, Jun. 1988.
- [30] J. V. Albro, G. A. Sohl, J. E. Bobrow, and F. C. Park, “On the computation of optimal high-dives,” in *IEEE International Conference on Robotics and Automation*, vol. 4, 2000, pp. 3958–3963.
- [31] M. F. Cohen, “Interactive spacetime control for animation,” *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 293–302, Jul. 1992.
- [32] L. Crawford, “Learning control of complex skills,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M98/53, 1998. [Online]. Available: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/1998/3500.html>
- [33] A. C. Fang and N. S. Pollard, “Efficient synthesis of physically valid human motion,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 417–426, Jul. 2003.
- [34] A. Safonova, J. K. Hodgins, and N. S. Pollard, “Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 514–521, Aug. 2004.
- [35] A. Sulejmanpašić and J. Popović, “Adaptation of performed ballistic motion,” *ACM Trans. Graph.*, vol. 24, no. 1, pp. 165–179, Jan. 2005.
- [36] K. Wampler and Z. Popović, “Optimal gait and form for animal locomotion,” *ACM Trans. Graph.*, vol. 28, no. 3, pp. 60:1–60:8, Jul. 2009.
- [37] X. Wei, J. Min, and J. Chai, “Physically valid statistical models for human motion generation,” *ACM Transactions on Graphics*, vol. 30, no. 3, pp. 1–10, May 2011.
- [38] K. Wampler, Z. Popović, and J. Popović, “Generalizing locomotion style to new animals with inverse optimal regression,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 49:1–49:11, Jul. 2014.
- [39] Z. Popović and A. Witkin, “Physically based motion transformation,” in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’99, 1999, pp. 11–20.
- [40] M. H. Raibert and J. K. Hodgins, “Animation of dynamic legged locomotion,” *SIGGRAPH Comput. Graph.*, vol. 25, no. 4, pp. 349–358, Jul. 1991.
- [41] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O’Brien, “Animating human athletics,” in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’95, ACM, 1995, pp. 71–78.
- [42] S. Coros, A. Karpathy, B. Jones, L. Reveret, and M. van de Panne, “Locomotion skills for simulated quadrupeds,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 59:1–59:12, Jul. 2011.
- [43] M. Hirofumi and I. Shimoyama, “Dynamic walk of a biped,” *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 60–74, 1994.

- [44] J. K. Hodgins and N. S. Pollard, "Adapting simulated behaviors for new characters," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '97, 1997, pp. 153–162.
- [45] A. D. Kuo, "Stabilization of lateral motion in passive dynamic walking," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 917–930, 1999.
- [46] S. Kudoh, T. Komura, and K. Ikeuchi, "Stepping motion for a human-like character to maintain balance against large perturbations," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2661–2666.
- [47] T. Niwa, S. Inagaki, and T. Suzuki, "Locomotion control of multi-legged robot based on follow-the-contact-point gait," in *2009 ICCAS-SICE*, 2009, pp. 2247–2253.
- [48] Y. Li, B. Li, J. Ruan, and X. Rong, "Research of mammal bionic quadruped robots: A review," in *2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*, Sept 2011, pp. 166–171.
- [49] K. Yin, K. Loken, and M. van de Panne, "Simbicon: Simple biped locomotion control," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.
- [50] L. Liu, K. Yin, M. van de Panne, T. Shao, and W. Xu, "Sampling-based contact-rich motion control," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 128:1–128:10, Jul. 2010.
- [51] S. Ha, Y. Ye, and C. K. Liu, "Falling and landing motion control for character animation," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 155:1–155:9, Nov. 2012.
- [52] L. Liu, K. Yin, M. van de Panne, and B. Guo, "Terrain runner: Control, parameterization, composition, and planning for highly dynamic motions," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 154:1–154:10, Nov. 2012.
- [53] M. da Silva, Y. Abe, and J. Popović, "Interactive simulation of stylized human locomotion," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 82:1–82:10, Aug. 2008.
- [54] U. Muico, Y. Lee, J. Popović, and Z. Popović, "Contact-aware nonlinear control of dynamic characters," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 81:1–81:9, Jul. 2009.
- [55] S. Jain and C. K. Liu, "Controlling physics-based characters using soft contacts," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 163:1–163:10, Dec. 2011.
- [56] J. Kim and N. S. Pollard, "Fast simulation of skeleton-driven deformable body characters," *ACM Trans. Graph.*, vol. 30, no. 5, pp. 121:1–121:19, Oct. 2011.
- [57] T. Shinar, C. Schroeder, and R. Fedkiw, "Two-way coupling of rigid and deformable bodies," in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '08, 2008, pp. 95–103.
- [58] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *2013 IEEE International Conference on Robotics and Automation, May 2013*, pp. 3982–3987.
- [59] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2550–2555.
- [60] C. Duriez and T. Bieze, "Soft robot modeling, simulation and control in real-time," in *Soft Robotics: Trends, Applications and Challenges*, 2017, pp. 103–109.
- [61] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, and C. Duriez, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.
- [62] A. K. Z. Z. R. M. Thor Morales Bieze, Frederick Largilliere and C. Duriez, "Finite element method-based kinematics and closed-loop control of soft, continuum manipulators," *Soft Robotics*, vol. 5, no. 3, 2018.
- [63] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 205–214, 1987.
- [64] G. Irving, J. Teran, and R. Fedkiw, "Tetrahedral and hexahedral invertible finite elements," *Graphical Models*, vol. 68, no. 2, pp. 66–89, 2006.
- [65] G. Hirota, S. Fisher, and M. Lin, "Simulation of non-penetrating elastic bodies using distance fields," Chapel Hill, NC, USA, Tech. Rep., 2000.
- [66] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable real-time deformations," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2002, pp. 49–54.
- [67] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," vol. 25, no. 4, pp. 809–836, 2006.
- [68] J. Barbič, M. da Silva, and J. Popović, "Deformable object animation using reduced optimal control," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 53:1–53:9, Jul. 2009.
- [69] J. Barbič, F. Sin, and E. Grinspun, "Interactive editing of deformable simulations," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 70:1–70:8, Jul. 2012.
- [70] S. Li, J. Huang, F. de Goes, X. Jin, H. Bao, and M. Desbrun, "Space-time editing of elastic motion through material optimization and reduction," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 108:1–108:10, Jul. 2014.
- [71] Z. Pan and D. Manocha, "Active animations of reduced deformable models with environment interactions," *ACM Trans. Graph.*, vol. 37, no. 3, pp. 36:1–36:17, Aug. 2018. [Online]. Available: <http://doi.acm.org/10.1145/3197565>
- [72] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popović, "Interactive skeleton-driven dynamic deformations," in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 586–593.
- [73] S.-H. Lee, E. Sifakis, and D. Terzopoulos, "Comprehensive biomechanical modeling and simulation of the upper body," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 4, p. 99, 2009.
- [74] L. Liu, K. Yin, B. Wang, and B. Guo, "Simulation and control of skeleton-driven soft body characters," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 215, 2013.
- [75] R. Ogden, "Large deformation isotropic elasticity-on the correlation of theory and experiment for incompressible rubberlike solids," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 326, no. 1567. The Royal Society, 1972, pp. 565–584.
- [76] M. Pozzi, E. Miguel, R. Deimel, M. Malvezzi, B. Bickel, O. Brock, and D. Prattichizzo, "Efficient fem-based simulation of soft robots modeled as kinematic chains," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–8.
- [77] M. Moore and J. Wilhelms, "Collision detection and response for computer animation," in *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '88, 1988, pp. 289–298.
- [78] R. Bridson, R. Fedkiw, and J. Anderson, "Robust treatment of collisions, contact and friction for cloth animation," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 594–603, Jul. 2002.
- [79] M. Bro-Nielsen and S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," *Computer Graphics Forum*, vol. 15, no. 3, pp. 57–66, 1996.
- [80] Y. Teng, M. Meyer, T. DeRose, and T. Kim, "Subspace condensation: Full space adaptivity for subspace deformations," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 76:1–76:9, Jul. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2766904>
- [81] M. Anitescu and F. A. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dynamics*, vol. 14, no. 3, pp. 231–247, 1997.
- [82] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton extraction by mesh contraction," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 44:1–44:10, Aug. 2008.
- [83] T.-C. Huang, Y.-J. Huang, and W.-C. Lin, "Real-time horse gait synthesis," *Computer Animation and Virtual Worlds*, vol. 24, no. 2, pp. 87–95, 2013.
- [84] A. A. Shabana, *Dynamics of multibody systems*. Cambridge university press, 2013.
- [85] C. K. Liu and S. Jain, "A quick tutorial on multibody dynamics," *Online tutorial*, June, p. 7, 2012.
- [86] K.-J. Bathe, *Finite element method*. Wiley Online Library, 2008.
- [87] D. Baraff and A. Witkin, "Large steps in cloth simulation," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 43–54.
- [88] P. Song, J.-S. Pang, and V. Kumar, "A semi-implicit time-stepping model for frictional compliant contact problems," *International Journal for Numerical Methods in Engineering*, vol. 60, no. 13, pp. 2231–2261, 2004.



Xudong Feng Xudong Feng is a Ph.D. candidate in the State Key Lab of CAD & CG, College of Computer Science, Zhejiang University. He received his bachelor degree in Tianjin University. His research interests include physical based simulation and animation, digital fabrication and reinforcement learning.

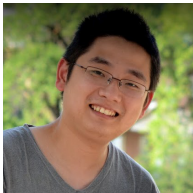


Jiafeng Liu is currently working toward the Master degree in computer engineering with the Zhejiang University. He received the bachelor's degree from the Hefei University of Technology in 2018. His research interests include character animation, robotics locomotion, machine learning and related topics.



Huamin Wang received the BEng degree from Zhejiang University, the MS degree from Stanford University, and the PhD degree in computer science from the Georgia Institute of Technology, in 2002, 2004 and 2009. He is an associate professor in the Department of Computer Science and Engineering, the Ohio State University. Before joining Ohio State University, he was a postdoctoral researcher in the Department of Electrical Engineering and Computer Sciences, the University of California, Berkeley. He is a

member of the IEEE.



Yin Yang received the PhD degree in computer science from the University of Texas at Dallas, in 2013. He is an assistant professor in the Department of Electrical Computer Engineering, the University of New Mexico, Albuquerque. His research interests include physics-based animation/ simulation and related applications, scientific visualization, and medical imaging analysis. He is a member of the IEEE.



Hujun Bao is a Chenkong professor in State Key Lab of CAD&CG, College of Computer Science at Zhejiang University. He received PhD degree in applied mathematics from Zhejiang university in 1993. His main research interests are computer graphics and computer vision, including real-time rendering technique, geometry computing, virtual reality, and 3D reconstruction, and has published more than 100 papers on prestigious academic journals and international conferences.



Bernd Bickel is an Assistant Professor, heading the Computer Graphics and Digital Fabrication group at IST Austria. He received Master and Ph.D degree in Computer Science from ETH Zurich. His research interests include computer graphics and its overlap into robotics, computer vision, biomechanics, material science, and digital fabrication. He received SIGGRAPH significant researcher award in 2017.



Weiwei Xu is a researcher with the State Key Lab of CAD & CG, College of Computer Science, Zhejiang University, awardee of NSFC Excellent Young Scholars Program in 2013. His main research interests include the digital geometry processing, physical simulation, computer vision and virtual reality. He has published around 70 papers on international graphics journals and conferences, including 16 papers on ACM TOG. He is a member of the IEEE.